

ETSI GS CIM 013 V2.1.1 (2024-04)



Context Information Management (CIM); NGSI-LD Test Purposes Descriptions

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

Reference

RGS/CIM-0013v211

Keywords

API, IoT, NGSI-LD, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
<https://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:
<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2024.
All rights reserved.

Contents

Intellectual Property Rights	5
Foreword.....	5
Modal verbs terminology.....	5
1 Scope	6
2 References	6
2.1 Normative references	6
2.2 Informative references.....	6
3 Definition of terms, symbols and abbreviations.....	7
3.1 Terms.....	7
3.2 Symbols.....	7
3.3 Abbreviations	7
4 Test Purposes Descriptions	8
4.1 jsonldContext	8
4.1.1 Consumption.....	8
4.1.1.1 ServeContext.....	8
4.1.1.2 ListContexts	13
4.1.2 Provision.....	30
4.1.2.1 AddContext	30
4.1.2.2 DeleteContext.....	33
4.2 CommonBehaviours.....	39
4.2.1 CommonResponses.....	39
4.2.1.1 VerifyNotAcceptableMediaType	39
4.2.1.2 VerifyLdContextNotAvailable.....	42
4.2.1.3 VerifyGETWithoutAccept	44
4.2.1.4 VerifyUnsupportedMediaType	46
4.2.1.5 VerifyMergePatchJson	49
4.3 ContextSource	50
4.3.1 Registration.....	50
4.3.1.1 DeleteContextSourceRegistration	50
4.3.1.2 UpdateContextSourceRegistration	52
4.3.1.3 RegisterContextSource.....	56
4.3.2 RegistrationSubscription.....	65
4.3.2.1 RetrieveContextSourceRegistrationSubscription	65
4.3.2.2 ContextSourceRegistrationSubscriptionNotificationBehaviour	67
4.3.2.3 DeleteContextSourceRegistrationSubscription	78
4.3.2.4 CreateContextSourceRegistrationSubscription	80
4.3.2.5 UpdateContextSourceRegistrationSubscription	87
4.3.2.6 QueryContextSourceRegistrationSubscriptions	91
4.3.3 Discovery.....	96
4.3.3.1 RetrieveContextSourceRegistration	96
4.3.3.2 QueryContextSourceRegistrations	99
4.4 ContextInformation	111
4.4.1 Consumption.....	111
4.4.1.1 TemporalEntity	111
4.4.1.1.1 RetrieveTemporalEvolutionOfEntity	111
4.4.1.1.2 QueryTemporalEvolutionOfEntities.....	124
4.4.1.2 Discovery	136
4.4.1.2.1 RetrieveAvailableEntityTypes.....	136
4.4.1.2.2 RetrieveAvailableAttributes	136
4.4.1.2.3 RetrieveDetailsOfAvailableEntityTypes	137
4.4.1.2.4 RetrieveAvailableAttributeInformation.....	138
4.4.1.2.5 RetrieveDetailsOfAvailableAttributes.....	139
4.4.1.2.6 RetrieveAvailableEntityTypeInformation	140
4.4.1.3 Entity.....	141

4.4.1.3.1	QueryEntities	141
4.4.1.3.2	RetrieveEntity	158
4.4.2	Subscription	167
4.4.2.1	UpdateSubscription	167
4.4.2.2	RetrieveSubscription	174
4.4.2.3	SubscriptionNotificationBehaviour	175
4.4.2.4	QuerySubscriptions	187
4.4.2.5	DeleteSubscription	190
4.4.2.6	CreateSubscription	191
4.4.3	Provision	195
4.4.3.1	TemporalEntity	195
4.4.3.1.1	UpdateTemporalRepresentationOfEntity	195
4.4.3.1.2	DeleteTemporalRepresentationOfEntity	196
4.4.3.1.3	CreateTemporalRepresentationOfEntity	198
4.4.3.2	BatchEntities	201
4.4.3.2.1	CreateBatchOfEntities	201
4.4.3.2.2	DeleteBatchOfEntities	209
4.4.3.2.3	UpsertBatchOfEntities	212
4.4.3.2.4	UpdateBatchOfEntities	218
4.4.3.3	EntityAttributes	222
4.4.3.3.1	DeleteEntityAttribute	222
4.4.3.3.2	PartialAttributeUpdate	227
4.4.3.3.3	UpdateEntityAttributes	232
4.4.3.3.4	AppendEntityAttributes	237
4.4.3.4	Entities	243
4.4.3.4.1	CreateEntity	243
4.4.3.4.2	DeleteEntity	252
4.4.3.5	TemporalEntityAttributes	254
4.4.3.5.1	DeleteAttribute	254
4.4.3.5.2	DeleteAttributeInstance	259
4.4.3.5.3	AddAttributes	264
4.4.3.5.4	PartialUpdateAttributeInstance	267
Annex A (informative):	Bibliography	273
Annex B (informative):	Change History	274
History		275

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document contains the description of each abstract test case using the Test Template [i.2] and using the Test Purposes Description Language identified in ETSI GS CIM 012 [2] and ETSI GR CIM 011 [i.1]. The test cases are described in tabular form. All clauses mentioned in tables of clause 4 of the present document are references to the NGSI-LD specification, hence to be found in [1].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <https://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

- [1] [ETSI GS CIM 009 \(V1.5.1\) \(2021-11\)](#): "Context Information Management (CIM); NGSI-LD API".
- [2] [ETSI GS CIM 012 \(V1.1.1\) \(2021-03\)](#): "Context Information Management (CIM); NGSI-LD Test Suite Structure".
- [3] Void.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI GR CIM 011 (V1.1.1) (2021-04): "Context Information Management (CIM); NGSI-LD Testing Framework: Test Purposes Description Language (TPDL)".
- [i.2] ETSI GS CIM 016 (V1.1.1) (2021-04): "Context Information Management (CIM); NGSI-LD Testing Framework: Test Template".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

NOTE: The letters "NGSI-LD" were added to most terms to confirm that they are distinct from other terms of similar/same name in use in other organizations, however, in the present document the letters "NGSI-LD" are generally omitted for brevity.

NGSI-LD Central Broker: NGSI-LD Context Broker that only uses a local storage when serving NGSI-LD requests, without involving any external Context Sources

NGSI-LD Context Broker: architectural component that implements all the NGSI-LD interfaces

NGSI-LD Context Consumer: agent that uses the query and subscription functionality of NGSI-LD to retrieve context information

NGSI-LD Context Producer: agent that uses the NGSI-LD context provision and/or registration functionality to provide or announce the availability of its context information to an NGSI-LD Context Broker

NGSI-LD Context Registry: software functional element where Context Sources register the information that they can provide

NOTE: It is used by Distribution Brokers and Federation Brokers to find the appropriate Context Sources which can provide the information required for serving an NGSI-LD request.

NGSI-LD Context Source: source of context information which implements the NGSI-LD consumption and subscription (and possibly provision) interfaces defined by the present document

NOTE: It is usually registered with an NGSI-LD Registry so that it can announce what kind of information it can provide, when requested, to Context Consumers and Brokers.

NGSI-LD Distribution Broker: NGSI-LD Context Broker that uses both local context information and registration information from an NGSI-LD Context Registry, to access matching context information from a set of distributed Context Sources

NGSI-LD Federation Broker: Distribution Broker that federates information from multiple underlying NGSI-LD Context Brokers and across domains

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CF	Config Identifier
DEL	DELETE
HTTP	HperText Transfer Protocol
JSON	Java Script Object Notation
PICS	Profile Implementation Conformance Statement
SUT	System Under Test
TP	Test Purpose
URI	Unified Resource Identification
URL	Unified Resource Location

4 Test Purposes Descriptions

4.1 jsonldContext

4.1.1 Consumption

4.1.1.1 ServeContext

TP Id	TP/NGSI-LD/CTX/Cons/053_06	
Test objective	Check that the numberOfHits is increased after using a Cached context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Cached @context created from a entity creation through downloading from external server. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_06_01	
Name	053_06_01 Check that the numberOfHits is increased after using a Cached context	
Parent Release	v1.5.1	
Description	Check that the numberOfHits is increased after using a Cached context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Check that the numberOfHits in the response body is set to '2' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/053_04	
Test objective	Check that one can get an increase numberOfHits after creation of a Hosted @context and using it	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Cached @context created from a entity creation through downloading from external server. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_04_01	
Name	053_04_01 Check that the numberOfHits is increased after using a Hosted context	
Parent Release	v1.5.1	
Description	Check that the numberOfHits is increased after using a Hosted context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'true' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Check that the numberOfHits in the response body is set to '2' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CTX/Cons/053_02	
Test objective	Check that one can serve a previous created @context with details equal to True	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_02_01	
Name	053_02_01 Serve a @context with details set to True	
Parent Release	v1.5.1	
Description	Check that one can serve a @context with details	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Response reason set to 'OK' and Serve a @context with Response Header: Content- Type set to application/ json and Serve a @context with Check the Body of the response should contain a @context with 'URL' key not Empty and 'string' type, with 'localId' key not Empty and 'string' type, with 'kind' key not Empty, 'string' type, and value set to 'Hosted', and 'timestamp' and 'DateTime' format and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'lastUsage' and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'numberOfHits' and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'extraInfo' and Serve a @context with Check that the only allowed keys in the response body of a @context are 'URL', 'localId', 'kind', 'timestamp', 'lastUsage', 'numberOfHits', 'extraInfo' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/053_05	
Test objective	Check that the context served by a context server is still in the broker after a ERROR_TYPE_LD_CONTEXT_NOT_AVAILABLE with details=true	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Cached @context created from a entity creation through downloading from external server. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_05_01	
Name	053_05_01 Check that the context served by a context server is still in the broker after a ERROR_TYPE_LD_CONTEXT_NOT_AVAILABLE with details=true	
Parent Release	v1.5.1	

Description	Check that the context served by a context server is still in the broker after a ERROR_TYPE_LD_CONTEXT_NOT_AVAILABLE with details=true	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Response reason set to 'OK' and Serve a @context with Response Header: Content- Type set to application/ json and Serve a @context with Check the Body of the response should contain a @context with 'URL' key not Empty and 'string' type, with 'localId' key not Empty and 'string' type, with 'kind' key not Empty, 'string' type, and value set to 'Cached', and 'timestamp' and 'DateTime' format and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'lastUsage' and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'numberOfHits' and Serve a @context with The Response Body '\$ {response.json()}' might contain the key 'extraInfo' and Serve a @context with Check that the only allowed keys in the response body of a @context are 'URL', 'localId', 'kind', 'timestamp', 'lastUsage', 'numberOfHits', 'extraInfo' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/053_07	
Test objective	Check that one can serve a ImplicitlyCreated @context with details set to true	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a ImplicitlyCreated @context created from a subscription query. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_07_01	
Name	053_07_01 Check that one can serve a ImplicitlyCreated @context with details set to true	
Parent Release	v1.5.1	
Description	Check that one can serve a ImplicitlyCreated @context with details set to true	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {implicit_id}' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Response reason set to 'OK' and Serve a @context with Response Header: Content- Type set to application/ json and Serve a @context with Check the Body of the response should contain a @context with 'URL' key not Empty and 'string' type, with 'localId' key not Empty and 'string' type, with 'kind' key not Empty, 'string' type, and value set to 'ImplicitlyCreated', and 'timestamp' and 'DateTime' format and Serve a @context with The Response Body '\$ </pre>	SUT -> Client

	<pre> {response.json()}' might contain the key 'lastUsage' and Serve a @context with The Response Body ` \$ {response.json()}' might contain the key 'numberOfHits' and Serve a @context with The Response Body ` \$ {response.json()}' might contain the key 'extraInfo' and Serve a @context with Check that the only allowed keys in the response body of a @context are 'URL', 'localId', 'kind', 'timestamp', 'lastUsage', 'numberOfHits', 'extraInfo' } </pre>	
--	--	--

TP Id	TP/NGSI-LD/CTX/Cons/053_03	
Test objective	Check that one gets an error when try to serve a @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_03_01	
Name	053_03_01 Serve a @contexts with a wrong id and correct details	
Parent Release	v1.5.1	
Description	Check that an error is returned when one requests for a @context that does not exist	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID 'wrong_id_context' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 404 and Serve a @context with Response reason set to 'Not Found' and Serve a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_03_02	
Name	053_03_02 Serve a @contexts with a valid id and incorrect details	
Parent Release	v1.5.1	
Description	Check that an error is returned when one requests for a @context that does not exist	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'other' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 400 and Serve a @context with Response reason set to 'Bad Request' and Serve a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} } </pre>	SUT -> Client

	and Response Boty Title is not equal to Response Body Detail }	
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_03_03	
Name	053_03_03 Serve a Cached @contexts with details set to false	
Parent Release	v1.5.1	
Description	Check that an error is returned when one requests for a @context that does not exist	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context- v1.6.jsonld' and details 'false' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 422 and Serve a @context with Response reason set to 'Unprocessable Content' and Serve a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ OperationNotSupported' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/053_01	
Test objective	Check that one can serve a previous created @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_01_01	
Name	053_01_01 Serve a @context without details	
Parent Release	v1.5.1	
Description	Check that one can serve a @context with details equal to empty or false	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details '\${EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Response reason set to 'OK' and Serve a @context with Response Header: Content- Type set to application/ json and Serve a @context with Check the Body of the response should contain the @context `@context- minimal-valid.json' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_01_02	
Name	053_01_02 Serve a @context with details equal to false	
Parent Release	v1.5.1	

Description	Check that one can serve a @context with details equal to empty or false	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {uri}' and details 'false' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Response reason set to 'OK' and Serve a @context with Response Header: Content- Type set to application/ json and Serve a @context with Check the Body of the response should contain the @context '@context- minimal-valid.json' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/053_08	
Test objective	Check that the numberOfHits is increased after using a ImplicitlyCreated context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.4	
Config Id	CF_01	
PICS Selection	PICS_SerOneSpeUse@co	
Initial conditions	with { the SUT containing a ImplicitlyCreated @context created from a subscription query. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/053_08_01	
Name	053_08_01 Check that the numberOfHits is increased after using a ImplicitlyCreated context	
Parent Release	v1.5.1	
Description	Check that the numberOfHits is increased after using a ImplicitlyCreated context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID '\$ {implicit_id}' and details 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Serve a @context with Response Status Code set to 200 and Serve a @context with Check that the numberOfHits in the response body is set to '2' } </pre>	SUT -> Client

4.1.1.2 ListContexts

TP Id	TP/NGSI-LD/CTX/Cons/052_06	
Test objective	Check that one can list all the @context available in the broker with no previous add @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAllCac@co	
Initial conditions	with { the SUT containing a ImplicitlyCreated @context created from a subscription query. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_01	
Name	052_06_01 List @contexts with neither details or kind and a created ImplicitlyCreated @context	

Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '4' and Query Parameter: list_contexts set to 'list_contexts' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_02	
Name	052_06_02 List @contexts with no details and kind set to hosted and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'list_contexts' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_03	
Name	052_06_03 List @contexts with no details and kind set to cached and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Cached' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and </pre>	SUT -> Client

	Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '3' and Query Parameter: list_contexts set to 'list_contexts' }	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_04	
Name	052_06_04 List @contexts with no details and kind set to implicitlycreated and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'list_contexts' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_05	
Name	052_06_05 List @contexts with details set to false and no kind and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to '\${EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '4' and Query Parameter: list_contexts set to 'list_contexts' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_06	
Name	052_06_06 List @contexts with details set to false and kind equal to hosted and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and }	SUT <- Client

	<pre> Query Parameter: kind set to 'Hosted' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'list_contexts' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_07	
Name	052_06_07 List @contexts with details set to false and kind equal to cached and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Cached' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '3' and Query Parameter: list_contexts set to 'list_contexts' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_06_08	
Name	052_06_08 List @contexts with details set to false and kind equal to implicitlycreated and a created ImplicitlyCreated @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'list_contexts' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/052_04	
Test objective	Check that one gets an error when try to list @context with wrong details or kind	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAllCac@co	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_04_01	
Name	052_04_01 List @contexts with no details and kind set to other and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'other' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 400 and List @contexts with Response reason set to 'Bad Request' and List @contexts with Response Body containing the type 'https://uri.etsi.org/ngsi-lid/errors/BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_04_02	
Name	052_04_02 List @contexts with details set to other and no kind and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'other' and Query Parameter: kind set to '\${EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 400 and List @contexts with Response reason set to 'Bad Request' and List @contexts with Response Body containing the type 'https://uri.etsi.org/ngsi-lid/errors/BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_04_03	
Name	052_04_03 List @contexts with details set to other and kind set to other and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'other' and Query Parameter: kind set to 'other' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 400 and List @contexts with Response reason set to 'Bad Request' and List @contexts with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_04_04	
Name	052_04_04 List @contexts with details set to true and kind set to other and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to 'other' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 400 and List @contexts with Response reason set to 'Bad Request' and List @contexts with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_04_05	
Name	052_04_05 List @contexts with details set to other and kind set to Hosted and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'other' and Query Parameter: kind set to 'Hosted' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 400 and List @contexts with Response reason set to 'Bad Request' and	SUT -> Client

	<pre>List @contexts with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }</pre>	
--	---	--

TP Id	TP/NGSI-LD/CTX/Cons/052_02	
Test objective	Check that one can list all the @context available in the broker with one add @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAllCac@co	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_01	
Name	052_02_01 List @contexts with neither details or kind and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to '\${EMPTY}' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '2' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to '\$ \${EMPTY}' }</pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_02	
Name	052_02_02 List @contexts with no details and kind equal to hosted and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Hosted' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and </pre>	SUT -> Client

	Query Parameter: kind set to 'Hosted' }	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_03	
Name	052_02_03 List @contexts with no details and kind equal to cached and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Cached' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Cached' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_04	
Name	052_02_04 List @contexts with no details and kind equal to implicitlycreated and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_05	
Name	052_02_05 List @contexts with details equal to false and no kind and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and	SUT <- Client

	<pre> Query Parameter: kind set to '\${EMPTY}' } then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '2' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to '\$ {EMPTY}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_06	
Name	052_02_06 List @contexts with details equal to false and kind equal to hosted and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Hosted' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_07	
Name	052_02_07 List @contexts with details equal to false and kind equal to cached abd with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Cached' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Cached' } </pre>	SUT -> Client

TP Permutation id	TP/NGSI-LD/CTX/Cons/052_02_08	
Name	052_02_08 List @contexts with details equal to false and kind equal to implicitlycreated and with previously one add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/052_05	
Test objective	Check that one can list all the @context available in the broker with several add @contexts with details equal to true	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAIIcAc@co	
Initial conditions	with { the SUT containing a set of three Hosted @contexts and the default Cached Core Context. }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_05_01	
Name	052_05_01 List @contexts with details set to true and no kind and with previously several add @contexts	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Response Body containing a Context element containing JSONObject with details of the contexts: * the expected length of contexts set to '4' * the list of contexts URI set to 'list_contexts=\$ {uris}' * response body to be checked set to 'response=\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_05_02	
Name	052_05_02 List @contexts with details set to true and kind set to hosted and with previously several add @contexts	

Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Response Body containing a Context element containing JSONObject with details of the contexts: * the expected length of contexts set to '3' * the list of contexts URI set to 'list_contexts=\$ {uris}' * response body to be checked set to 'response=\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_05_03	
Name	052_05_03 List @contexts with details set to true and kind set to cached abd with previously several add @contexts	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to 'Cached' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Response Body containing a Context element containing JSONObject with details of the contexts: * the expected length of contexts set to '1' * the list of contexts URI set to 'list_contexts=\$ {uris}' * response body to be checked set to 'response=\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_05_04	
Name	052_05_04 List @contexts with details set to true and kind set to implicitlycreated and with previously several add @contexts	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to </pre>	SUT -> Client

	<pre> 200 and List @contexts with Response reason set to 'OK' and List @contexts with Response Body containing a Context element containing JSONObject with details of the contexts: * the expected length of contexts set to '0' * the list of contexts URI set to 'list_contexts=\$ {uris}' * response body to be checked set to 'response=\$ {response.json()}' } </pre>	
--	---	--

TP Id	TP/NGSI-LD/CTX/Cons/052_03	
Test objective	Check that one can list all the @context available in the broker with several add @contexts	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAllCac@co	
Initial conditions	with { the SUT containing a set of three Hosted @contexts and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_01	
Name	052_03_01 List @contexts with neither details or kind and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '4' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to '\$ {EMPTY}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_02	
Name	052_03_02 List @contexts with no details and kind equal to hosted and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of </pre>	SUT -> Client

	identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '3' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Hosted' }	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_03	
Name	052_03_03 List @contexts with no details and kind equal to cached and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Cached' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Cached' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_04	
Name	052_03_04 List @contexts with no details and kind equal to implicitlycreated and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_05	
Name	052_03_05 List @contexts with details equal to false and no kind and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to '\${EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '4' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to '\$ {EMPTY}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_06	
Name	052_03_06 List @contexts with details equal to false and kind equal to hosted and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Hosted' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '3' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Hosted' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_07	
Name	052_03_07 List @contexts with details equal to false and kind equal to cached abd with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Cached' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'Cached' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_03_08	
Name	052_03_08 List @contexts with details equal to false and kind equal to implicitlycreated and with previously several add @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' and Query Parameter: list_contexts set to 'uri_list' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Cons/052_01	
Test objective	Check that one can list all the @context available in the broker with no previous add @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.3	
Config Id	CF_01	
PICS Selection	PICS_LisAllCac@co	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_01	
Name	052_01_01 List @contexts with neither details or kind and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of </pre>	SUT -> Client

	<pre> identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' } </pre>	
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_02	
Name	052_01_02 List @contexts with no details and kind equal to hosted and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_03	
Name	052_01_03 List @contexts with no details and kind equal to cached and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'Cached' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_04	
Name	052_01_04 List @contexts with no details and kind equal to implicitlycreated and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to '\${EMPTY}' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_05	
Name	052_01_05 List @contexts with details equal to false and no kind and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_06	
Name	052_01_06 List @contexts with details equal to false and kind equal to hosted and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Hosted' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_07	
Name	052_01_07 List @contexts with details equal to false and kind equal to cached and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' </pre>	SUT <- Client

	<pre> method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'Cached' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '1' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Cons/052_01_08	
Name	052_01_08 List @contexts with details equal to false and kind equal to implicitlycreated and not previously created @context	
Parent Release	v1.5.1	
Description	Check that one can list @contexts	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'false' and Query Parameter: kind set to 'ImplicitlyCreated' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: List @contexts with Response Status Code set to 200 and List @contexts with Response reason set to 'OK' and List @contexts with Check Context Response Body Containing a list of identifiers and Query Parameter: response_body set to 'response.json()' and Query Parameter: expected_length set to '0' } </pre>	SUT -> Client

4.1.2 Provision

4.1.2.1 AddContext

TP Id	TP/NGSI-LD/CTX/Prov/050_02	
Test objective	Check that one receives a 400 Bad Request creating a @context if the content is incorrect	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.2	
Config Id	CF_01	
PICS Selection	PICS_AddUse@coTheInt	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_02_01	
Name	050_02_01 Checking Incorrect Payload	
Parent Release	v1.5.1	
Description	Check that the payload include "@context"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/jsonldContexts' method set to 'POST' Add a new @context based on the payload defined in the file @context- incorrect.json } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Add a new @context with Response Status Code set to 400 and Add a new @context with Response reason set to 'Bad Request' and Add a new @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_02_02	
Name	050_02_02 Checking Wrong JSON	
Parent Release	v1.5.1	
Description	Check that the payload include "@context"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts' method set to 'POST' Add a new @context based on the payload defined in the file @context- wrong-json.json } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Add a new @context with Response Status Code set to 400 and Add a new @context with Response reason set to 'Bad Request' and Add a new @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/050_04	
Test objective	Check that one can add a hosted @context with list of URIs and each of them are cached @cocontexts	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.2	
Config Id	CF_01	
PICS Selection	PICS_AddUse@coTheInt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_04_01	
Name	050_04_01 Add a valid hosted @context with URIs and check that the URIs are Cached @contexts	
Parent Release	v1.5.1	
Description	Check that one can add a @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts' method set to 'POST' Add a new @context based on the payload defined in the file @context- cached-valid.json } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Add a new @context with Response Status Code set to 201 and Add a new @context with Response does not contain a body and Add a new @context with Response reason set to 'Created' and Serve a @context with Check that the Kind of the @context is set to 'Hosted' and Notification with Check that for each @context in the response body, they are of the type 'Cached' } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CTX/Prov/050_01	
Test objective	Check that one can add a hosted @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.2	
Config Id	CF_01	
PICS Selection	PICS_AddUse@coTheInt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_01_01	
Name	050_01_01 Add a valid hosted @context from key=value	
Parent Release	v1.5.1	
Description	Check that one can add a @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts' method set to 'POST' Add a new @context based on the payload defined in the file @context- minimal-valid.json } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Add a new @context with Response Status Code set to 201 and Add a new @context with Response does not contain a body and Add a new @context with Response reason set to 'Created' and Serve a @context with Check that the Kind of the @context is set to 'Hosted' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_01_02	
Name	050_01_02 Add a valid cached @context from URI	
Parent Release	v1.5.1	
Description	Check that one can add a @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts' method set to 'POST' Add a new @context based on the payload defined in the file @context- cached-valid.json } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Add a new @context with Response Status Code set to 201 and Add a new @context with Response does not contain a body and Add a new @context with Response reason set to 'Created' and Serve a @context with Check that the Kind of the @context is set to 'Hosted' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/050_03	
Test objective	Check that one can create a implicitlycreated @context through creating a subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.2	
Config Id	CF_01	
PICS Selection	PICS_AddUse@coTheInt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/050_03_01	
Name	050_03_01 Check the creation of ImplicitelyCreted @context	
Parent Release	v1.5.1	
Description	Check that one can create a subscription	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts?\${params_as_string}' method set to 'GET' List @contexts: and Query Parameter: details set to 'true' and Query Parameter: kind set to 'ImplicitlyCreated' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 200 and Notification with Check that the Kind of the @context is set to 'ImplicitlyCreated' }	SUT -> Client

4.1.2.2 DeleteContext

TP Id	TP/NGSI-LD/CTX/Prov/051_09	
Test objective	Check that one gets an error when try to delete the core @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a core context and it has been deleted with reload set to true. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_09_01	
Name	051_09_01 Delete a core @contexts with reload set to true and check if the context has been download again by the broker	
Parent Release	v1.5.1	
Description	Delete a core @contexts with reload set to true and check downloaded core context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'GET' Serve a @context with contextID 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context- v1.6.jsonld' and details 'true' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Serve a @context with Response Status Code set to 200 }	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_08	
Test objective	Check that one gets an error when try to delete the core @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing an initial state }	

Permutations							
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_08_01						
Name	051_08_01 Delete a core @contexts and return an error with no reload						
Parent Release	v1.5.1						
Description	Check that one gets an error when try to delete the core @context						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } } </td> <td>SUT <- Client</td> </tr> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } } </td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT <- Client	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT -> Client
Test events	Direction						
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT <- Client						
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT -> Client						
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_08_02						
Name	051_08_02 Delete a core @contexts and return an error with reload set to false						
Parent Release	v1.5.1						
Description	Check that one gets an error when try to delete the core @context						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } } </td> <td>SUT <- Client</td> </tr> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } } </td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT <- Client	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT -> Client
Test events	Direction						
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT <- Client						
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'https:// uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld' } then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } }	SUT -> Client						

TP Id	TP/NGSI-LD/CTX/Prov/051_02
Test objective	Check that one gets an error when trying to delete an unknown @context identifier
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5
Config Id	CF_01
PICS Selection	PICS_DelOneSpe@colnt
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_02_01
Name	051_02_01 Delete a @context with unknown @context identifier
Parent Release	v1.5.1
Description	Check that an error message is obtained in the response when one tries to delete a @context with unknowwn id

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\$ {random_url}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 404 and Delete a @context with Response reason set to 'Not Found' and Delete a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_04	
Test objective	Check that one gets an error when try to delete a @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_04_01	
Name	051_04_01 Delete a @contexts with a wrong id and reload set to true	
Parent Release	v1.5.1	
Description	Check that one can delete a hosted @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'wrong_id_context' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 404 and Delete a @context with Response reason set to 'Not Found' and Delete a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_04_02	
Name	051_04_02 Delete a @contexts with a wrong id and reload set to false	
Parent Release	v1.5.1	
Description	Check that one can delete a hosted @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' } </pre>	SUT <- Client

	<pre> Delete a @context whose 'URI' set to 'wrong_id_context' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 404 and Delete a @context with Response reason set to 'Not Found' and Delete a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_04_03	
Name	051_04_03 Delete a @contexts with a wrong id and no reload value	
Parent Release	v1.5.1	
Description	Check that one can delete a hosted @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'wrong_id_context' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 404 and Delete a @context with Response reason set to 'Not Found' and Delete a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_04_04	
Name	051_04_04 Delete a @contexts with a wrong id and wrong reload value	
Parent Release	v1.5.1	
Description	Check that one can delete a hosted @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to 'wrong_id_context' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 404 and Delete a @context with Response reason set to 'Not Found' and Delete a @context with Response Body containing the type '\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_04_05	
Name	051_04_05 Delete a Hosted @contexts with a valid id and reload set to true	
Parent Release	v1.5.1	

Description	Check that one can delete a hosted @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\${uri}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_06	
Test objective	Delete a @context whose kind is ImplicitlyCreated without reload param	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a ImplicitlyCreated @context created from a subscription query. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_06_01	
Name	051_06_01 Delete a @context whose kind is ImplicitlyCreated without reload param	
Parent Release	v1.5.1	
Description	Check that one can delete a ImplicitlyCreated @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\$ {implicit_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 204 and Delete a @context with Response reason set to 'No Content' and Delete a @context with Response does not contain a body } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_01	
Test objective	Check that one can delete a previous created hosted @context without reload param	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a Hosted @context and the default Cached Core Context. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_01_01	
Name	051_01_01 Delete a @context whose kind is hosted without reload param	
Parent Release	v1.5.1	
Description	Check that one can delete a hosted @context	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\${uri}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 204 and Delete a @context with Response reason set to 'No Content' and Delete a @context with Response does not contain a body }	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_03	
Test objective	Check that one can delete a previous created cached @context without reload param	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a Cached @context added from a URL. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_03_01	
Name	051_03_01 Delete a @context whose kind is cached without reload param	
Parent Release	v1.5.1	
Description	Check that one can delete a cached @context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\$ {contextId}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 204 and Delete a @context with Response reason set to 'No Content' and Delete a @context with Response does not contain a body }	SUT -> Client

TP Id	TP/NGSI-LD/CTX/Prov/051_07	
Test objective	Check that one cannot delete a ImplicitlyCreated @context with reload set to true	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a ImplicitlyCreated @context created from a subscription query. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_07_01	
Name	051_07_01 Delete a ImplicitlyCreated @contexts with a valid id and reload set to true	
Parent Release	v1.5.1	
Description	Check that one cannot delete a ImplicitlyCreated @context with reload set to true	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\$ {implicit_id}' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 400 and Delete a @context with Response reason set to 'Bad Request' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CTX/Prov/051_05	
Test objective	Check that one gets an error if one created an entity with a context (Cached context) and one tries to delete it with reload=true	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.13.5	
Config Id	CF_01	
PICS Selection	PICS_DelOneSpe@colnt	
Initial conditions	with { the SUT containing a Cached @context created from a entity creation through downloading from external server. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CTX/Prov/051_05_01	
Name	051_05_01 Delete and Reload a Cached @context with no communication with the Context Server	
Parent Release	v1.5.1	
Description	Check that one gets an error if one tries to reload a cached context with no communication with the context server	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/jsonldContexts/{contextId}' method set to 'DELETE' Delete a @context whose 'URI' set to '\${uri}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete a @context with Response Status Code set to 503 and Delete a @context with Response reason set to 'Service Unavailable' and Delete a @context with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ LdContextNotAvailable' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

4.2 CommonBehaviours

4.2.1 CommonResponses

4.2.1.1 VerifyNotAcceptableMediaType

TP Id	TP/NGSI-LD/CB/HTTP/049_01
Test objective	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json"
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4
Config Id	CF_01

PICS Selection	PICS_HTTP_PRECOND	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_01_01	
Name	049_01_01 Endpoint get /entities/{entityId}	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json" (get /entities/{entityId})	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '{entity_id}' and Query Parameter: accept set to 'application/xml' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Query Entity with Response Status Code set to 406 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_01_02	
Name	049_01_02 Endpoint get /subscriptions/{subscriptionId}	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json" (get /subscriptions/{subscriptionId})	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '{id}' and Query Parameter: accept set to 'application/xml' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Subscription with Response Status Code set to 406 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_01_03	
Name	049_01_03 Endpoint get /csourceRegistrations/	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json" (get /csourceRegistrations/)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/csourceRegistrations/' method set to 'GET' Request a Context Source Registration with Return Entity Type set to 'Building' Header['Accept'] set to 'application/xml' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Query Context Source Registrations With Return with Response Status Code set to 406 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_01_04	
Name	049_01_04 Endpoint get /csourceSubscriptions/	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json" (get /csourceSubscriptions/)	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: accept set to 'application/xml' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Query Context Source Registration Subscriptions with Response Status Code set to 406 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_01_05	
Name	049_01_05 Endpoint get /temporal/entities	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header does not imply "application/json" nor "application/ld+json" (get /temporal/entities)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-08-01T12:05:00Z' and Query Parameter: accept set to 'application/xml' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Query Temporal Representation Of Entities with Response Status Code set to 406 }	SUT -> Client

TP Id	TP/NGSI-LD/CB/HTTP/049_02	
Test objective	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header is "application/geo+json" for operations different than "Retrieve Entity" and "Query Entity"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4	
Config Id	CF_01	
PICS Selection	PICS_HTTP_PRECOND	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_02_01	
Name	049_02_01 Retrieve subscription by id	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header is "application/geo+json" for operations different than "Retrieve Entity" and "Query Entity" (get /subscriptions/{subscriptionId})	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '\${id}' and Query Parameter: accept set to 'application/geo+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Subscription with Response Status Code set to 406 }	SUT -> Client

TP Permutation id	TP/NGSI-LD/CB/HTTP/049_02_02	
Name	049_02_02 Query temporal entities	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header is "application/geo+json" for operations different than "Retrieve Entity" and "Query Entity" (get /temporal/entities)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-08-01T12:05:00Z' and Query Parameter: accept set to 'application/geo+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Query Temporal Representation Of Entities with Response Status Code set to 406 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/049_02_03	
Name	049_02_03 Query context source registration	
Parent Release	v1.3.1	
Description	Verify throwing 406 HTTP status code (Not Acceptable Media Type) if the "Accept" header is "application/geo+json" for operations different than "Retrieve Entity" and "Query Entity" (get /csourceRegistrations)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/csourceRegistrations' method set to 'GET' Request a Context Source Registration with Return Entity Type set to 'Building' Header['Accept'] set to 'application/geo+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Query Context Source Registrations With Return with Response Status Code set to 406 } </pre>	SUT -> Client

4.2.1.2 VerifyLdContextNotAvailable

TP Id	TP/NGSI-LD/CB/HTTP/043_01	
Test objective	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.2.2	
Config Id	CF_01	
PICS Selection	PICS_IGNORESYSPROPS_AND_SYSPROPS_IFASKED	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/043_01_01	
Name	043_01_01 Create entity	
Parent Release	v1.3.1	
Description	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved (Create entity)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' </pre>	SUT <- Client

	<pre> ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building- unretrievable-context-sample.jsonld' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 503 and Create Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ LdContextNotAvailable' and Create Entity Selecting Content Type with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/043_01_02	
Name	043_01_02 Create subscription	
Parent Release	v1.3.1	
Description	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved (Create subscription)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/ method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscriptions/ subscription-unretrievable-context-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 503 and Create Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ LdContextNotAvailable' and Create Subscription with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/043_01_03	
Name	043_01_03 Create Temporal Representation of Entities	
Parent Release	v1.3.1	
Description	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved (Create Temporal Representation of Entities)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'bus- temporal-representation-unretrievable-context-sample.jsonld' and Query Parameter: content_type set to 'application/ ld+json' and Query Parameter: accept set to '\$ {EMPTY}' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 503 and Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ LdContextNotAvailable' and Create Or Update Temporal Representation Of Entity Selecting Content Type with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/043_01_04	
Name	043_01_04 Batch entity create	
Parent Release	v1.3.1	
Description	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved (Batch entity create)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload set to a list of entities to be created } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 503 and Batch Create Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ LdContextNotAvailable' and Batch Create Entities with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/043_01_05	
Name	043_01_05 Create context source registration	
Parent Release	v1.3.1	
Description	Verify receiving 503 - LdContextNotAvailable error if remote JSON-LD @context cannot be retrieved (Create context source registration)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Create Context Source Registration With Return with Response Status Code set to 503 } </pre>	SUT -> Client

4.2.1.3 VerifyGETWithoutAccept

TP Id	TP/NGSI-LD/CB/HTTP/045_01
Test objective	Verify that on a GET HTTP request if nothing is specified on the Accept header, "application/json" is assumed
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4
Config Id	CF_01
PICS Selection	PICS_HTTP_PRECOND

Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/045_01_01	
Name	045_01_01 Endpoint /entities/{entityId}	
Parent Release	v1.3.1	
Description	Verify that on a GET HTTP request if nothing is specified on the Accept header, "application/json" is assumed (/entities/{entityId})	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {id}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" and Query Parameter: accept set to '*/*' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 200 and Create Entity Selecting Content Type with Response Header: Content- Type set to application/ json and Notification with Response Header: Link is not Empty } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/045_01_02	
Name	045_01_02 Endpoint /subscriptions/{subscriptionId}	
Parent Release	v1.3.1	
Description	Verify that on a GET HTTP request if nothing is specified on the Accept header, "application/json" is assumed (/subscriptions/{subscriptionId})	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '\$ {id}' and Query Parameter: accept set to '*/*' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 201 and Create Subscription with Response Status Code set to 200 and Create Subscription with Response Header: Content- Type set to application/ json and Notification with Response Header: Link is not Empty } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/045_01_03	
Name	045_01_03 Endpoint /csourceRegistrations/	
Parent Release	v1.3.1	
Description	Verify that on a GET HTTP request if nothing is specified on the Accept header, "application/json" is assumed (/csourceRegistrations/)	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/csourceRegistrations' method set to 'GET' Request a Context Source Registration with Return Entity Type set to 'Vehicle' Header['Accept'] set to '*/*' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Create Context Source Registration With Return with Response Status Code set to 200 and Create Context Source Registration With Return with Response Header: Content- Type set to application/ json and Notification with Response Header: Link is not Empty }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/045_01_04	
Name	045_01_04 Endpoint /temporal/entities	
Parent Release	v1.3.1	
Description	Verify that on a GET HTTP request if nothing is specified on the Accept header, "application/json" is assumed (/temporal/entities)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/temporal/entities' method set to 'GET' Request a Temporal Representation of Entities with Return }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 200 and Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Header: Content- Type set to application/ json and Notification with Response Header: Link is not Empty }	SUT -> Client

4.2.1.4 VerifyUnsupportedMediaType

TP Id	TP/NGSI-LD/CB/HTTP/048_01
Test objective	Verify throwing 415 HTTP status code (Unsupported Media Type) if "Content-Type" header is not "application/json" or "application/ld+json"
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4
Config Id	CF_01
PICS Selection	PICS_HTTP_PRECOND
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CB/HTTP/048_01_01
Name	048_01_01 Endpoint patch /entities/{entityId}/attrs/{attrId}
Parent Release	v1.3.1
Description	Verify throwing 415 HTTP status code (Unsupported Media Type) if "Content-Type" header is not "application/json" or "application/ld+json" (patch /entities/{entityId}/attrs/{attrId})

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/subscriptions/' method set to 'POST' Request Header['Content-Type'] set to 'application/xml' and payload defined in file: 'csourceSubscriptions/ subscription-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Create Subscription with Response Status Code set to 415 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/048_01_05	
Name	048_01_05 Endpoint post /entityOperations/create	
Parent Release	v1.3.1	
Description	Verify throwing 415 HTTP status code (Unsupported Media Type) if "Content-Type" header is not "application/json" or "application/ld+json" (post /entityOperations/create)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/xml' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Batch Create Entities with Response Status Code set to 415 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CB/HTTP/048_01_06	
Name	048_01_06 Endpoint post /temporal/entities/	
Parent Release	v1.3.1	
Description	Verify throwing 415 HTTP status code (Unsupported Media Type) if "Content-Type" header is not "application/json" or "application/ld+json" (post /temporal/entities/)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'vehicle- temporal-representation-sample.jsonld' and Query Parameter: content_type set to 'application/ xml' and Query Parameter: accept set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 415 }	SUT -> Client

4.2.1.5 VerifyMergePatchJson

TP Id	TP/NGSI-LD/CB/HTTP/044_02	
Test objective	Verify that PATCH HTTP requests can be done with "application/merge-patch+json" as Content-Type	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4	
Config Id	CF_01	
PICS Selection	PICS_HTTP_PRECOND	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Subscription \${subscription} with an id set to \${subscription_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/044_02_01	
Name	044_02_01 Endpoint /subscriptions/{subscriptionId}	
Parent Release	v1.3.1	
Description	Verify that PATCH HTTP requests can be done with "application/merge-patch+json" as Content-Type	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Link'] contain the context 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Header['Content-Type'] set to 'application/merge- patch+json' and Payload defined in file 'subscriptions/fragments/ subscription-update-sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Update Subscription with Response Status Code set to 204 } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CB/HTTP/044_01	
Test objective	Verify that PATCH HTTP requests can be done with "application/merge-patch+json" as Content-Type	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.4	
Config Id	CF_01	
PICS Selection	PICS_HTTP_PRECOND	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CB/HTTP/044_01_01	
Name	044_01_01 endpoint /entities/{entityId}/attrs/{attrId}	
Parent Release	v1.3.1	
Description	Verify that PATCH HTTP requests can be done with "application/merge-patch+json" as Content-Type	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: AttributeId set to 'brandName' and Query Parameter: fragment_filename set to 'vehicle- brandname-fragment.json' and Query Parameter: content_type set to 'application/ merge-patch+json' and </pre>	SUT <- Client

	<pre> Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } then { the SUT sends a valid Response for the operation: Partial Update Entity Attributes with Response Status Code set to 204 } </pre>	
		SUT -> Client

4.3 ContextSource

4.3.1 Registration

4.3.1.1 DeleteContextSourceRegistration

TP Id	TP/NGSI-LD/CS/REG/035_03	
Test objective	Check that one cannot delete a context source registration by id if the id is not known to the system	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/035_03_01	
Name	035_03_01 Delete a context source registration by id	
Parent Release	v1.3.1	
Description	Check that one cannot delete a context source registration by id if the id is not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'DELETE' Delete Context Source Registration with id set to '\$ {registration_id}' } then { the SUT sends a valid Response for the operations: Delete Context Source Registration With Return with Response Status Code set to 404 and Delete Context Source Registration With Return with Response body containing 'title' element } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/035_01	
Test objective	Check that one can delete a context source registration by id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/035_01_01	
Name	035_01_01 Delete a context source registration by id	
Parent Release	v1.3.1	
Description	Check that one can delete a context source registration by id	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'DELETE' Delete Context Source Registration with id set to '\$ {registration_id}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Delete Context Source Registration With Return with Response Status Code set to 204 and Retrieve Context Source Registration with Response Status Code set to \${response1.status_code} }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/035_02	
Test objective	Check that one cannot delete a context source registration under some conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/035_02_01	
Name	035_02_01 Delete a Context Source Registration if the Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot delete a context source registration under some conditions	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'DELETE' Delete Context Source Registration with id set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Context Source Registration With Return with Response Status Code set to 405 and Delete Context Source Registration With Return with Response Body containing the type '\$ {EMPTY}' and Delete Context Source Registration With Return with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REG/035_02_02	
Name	035_02_02 Delete a Context Source Registration if the Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot delete a context source registration under some conditions	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'DELETE' Delete Context Source Registration with id set to 'invalidURI' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Context Source Registration With Return with Response Status Code set to 400 and Delete Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Delete Context Source Registration With Return with Response body containing 'title' element }	SUT -> Client

4.3.1.2 UpdateContextSourceRegistration

TP Id	TP/NGSI-LD/CS/REG/034_05	
Test objective	Check that one can update a context source registration by id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_05_01	
Name	034_05_01 Update a context source registration to never expire	
Parent Release	v1.3.1	
Description	Check that one can update a context source registration by id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to '\$ {registration_id}' and registration update fragment set to '\$ {registration_update_fragment}' and content- type set to 'application/json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 204 and Retrieve Context Source Registration with Check that response body does not contain the key '\$.expiresAt' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/034_03	
Test objective	Check that one cannot update a context source registration by id if the id is not known to the system	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT containing an initial state }	

Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_03_01	
Name	034_03_01 Update a context source registration by id if the id is not known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration by id if the id is not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to '\$ {registration_id}' and registration update fragment set to '\$ {fragment_with_id}' and content- type set to 'application/ld+json' } then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 404 and Update Context Source Registration With Return with Response reason set to 'Not Found' and Update Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/034_01	
Test objective	Check that one can update a context source registration by id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_01_01	
Name	034_01_01 Update a context source registration by id	
Parent Release	v1.3.1	
Description	Check that one can update a context source registration by id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to '\$ {registration_id}' and registration update fragment set to '\$ {registration_update_fragment}' and content- type set to 'application/json' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 204 and Retrieve Context Source Registration with Response Body containing the key '['endpoint']', with the value 'http:// my.new.csource.org:1026' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CS/REG/034_06	
Test objective	Check that one cannot update a context source registration if the Id is not present	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_06_01	
Name	034_06_01 Update a context source registration by id if the Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration under some conditions	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to '\$ {EMPTY}' and registration update fragment set to '\$ {fragment_with_id}' and content- type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 405 and Update Context Source Registration With Return with Response reason set to 'Method Not Allowed' and Update Context Source Registration With Return with Response does not contain a body } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/034_04	
Test objective	Check that one cannot update a context source registration under some conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_04_01	
Name	034_04_01 Update a context source registration if the request body is invalid	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration if the request body is invalid	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'PATCH' Update Context Source Registration with CSR Id set to '\$ </pre>	SUT <- Client

	{registration_id}' and subscription update from file 'context-source-registration-invalid-json-sample.json' }	
	then { the SUT sends a valid Response for the operations: Update Context Source Registration From File with Response Status Code set to 400 and Update Context Source Registration From File with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Update Context Source Registration From File with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REG/034_02	
Test objective	Check that one cannot update a context source registration under some conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REG/034_02_01	
Name	034_02_01 Update a context source registration by id if the Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration under some conditions	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to 'invalidURI' and registration update fragment set to '\$ {fragment_with_id}' and content-type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 400 and Update Context Source Registration With Return with Response reason set to 'Bad Request' and Update Context Source Registration With Return with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REG/034_02_02	
Name	034_02_02 Update a context source registration if the request body is not of the same data type	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration under some conditions	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}'	SUT <- Client

	<pre> method set to 'PATCH' Update Context Source Registration with id set to '\$ {valid_registration_id}' and registration update fragment set to '\$ {fragment_with_id}' and content- type set to 'application/ld+json' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 400 and Update Context Source Registration With Return with Response reason set to 'Bad Request' and Update Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REG/034_02_03	
Name	034_02_03 Update a context source registration if one attempts to remove a mandatory property	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration under some conditions	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{registration_id}' method set to 'PATCH' Update Context Source Registration with id set to '\$ {valid_registration_id}' and registration update fragment set to '\$ {fragment_with_id}' and content- type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration With Return with Response Status Code set to 400 and Update Context Source Registration With Return with Response reason set to 'Bad Request' and Update Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

4.3.1.3 RegisterContextSource

TP Id	TP/NGSI-LD/CS/CSR/033_02
Test objective	Check that one cannot create a context source with invalid content
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.2
Config Id	CF_05
PICS Selection	PICS_CsoRegCre
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CS/CSR/033_02_01
Name	033_02_01 Create a context source registration with invalid JSON file

Parent Release	v1.3.1	
Description	Create a context source registration with invalid JSON file	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content- Type set to 'application/ ld+json' and payload set to '\$ {subscription_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration with Response Status Code set to 400 and Create Context Source Registration with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Context Source Registration with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_01_02	
Test objective	Check that one can create a context source registration that never expires	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.2	
Config Id	CF_05	
PICS Selection	PICS_CsoRegCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_01_02	
Name	033_01_02 Create Context Source Registration That Never Expires	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration that never expires	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: '\$ {registration_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Create Context Source Registration With Return with Response Header: Location containing \$\$ {registration_id} and Retrieve Context Source Registration with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_06
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term

Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_06_01	
Name	033_06_01 Create one context source registration containing a JSON-LD @context with a JSON content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 400 and Create Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Context Source Registration With Return with Response body containing 'title' element } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_04	
Test objective	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_04_01	
Name	033_04_01 Create one context source registration using a provided Link header with JSON content type with Context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key '['information'] } </pre>	SUT <- Client
		SUT -> Client

	[0]['entities'][0] ['type'], with the value 'Building' }	
TP Permutation id	TP/NGSI-LD/CS/CSR/033_04_02	
Name	033_04_02 Create one context source registration using a provided Link header with JSON content type without Context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key '['information'] [0]['entities'][0] ['type'], with the value 'https://ngsi-ld-test-suite/context#Building' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_01_03	
Test objective	Check that one can create a context source registration without specifying an ID	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.2	
Config Id	CF_05	
PICS Selection	PICS_CsoRegCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_01_03	
Name	033_01_03 Create Context Source Registration Without Specifying an ID	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration without specifying an ID	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys }	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_03	
Test objective	Check that one cannot create a context source registration that already exists	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.2	
Config Id	CF_05	
PICS Selection	PICS_CsoRegCre	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_03_01	
Name	033_03_01 Create a context source registration that already exists	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration that already exists	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 409 and Create Context Source Registration With Return with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_01_01	
Test objective	Check that one can create a context source registration with specific ID and expiration date	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.9.2	
Config Id	CF_05	
PICS Selection	PICS_CsoRegCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_01_01	
Name	033_01_01 Create Context Source Registration With Specific Date Expiration Date	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration with specific ID and expiration date	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\$ {registration_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_05	
Test objective	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_05_01	
Name	033_05_01 Create one context source registration using the default context with JSON content type with Context	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context and retrieve the information with ngsild context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key ['information'] [0]['entities'][0]['type'], with the value 'ngsi-ld:default-context/Building' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/CSR/033_05_02	
Name	033_05_02 Create one context source registration using the default context with JSON content type without Context	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context and retrieve the information without ngsild context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key ['information'] [0]['entities'][0] ['type'], with the value 'Building' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_07
Test objective	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json"

Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_07_01	
Name	033_07_01 Create one context source registration using a JSON-LD @context obtained from the request payload with Context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and retrieve the information with ngsild context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key '["information"] [0]["entities"][0] ["type"]', with the value 'Vehicle' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/CSR/033_07_02	
Name	033_07_02 Create one context source registration using a JSON-LD @context obtained from the request payload without Context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and retrieve the information without ngsild context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 201 and Retrieve Context Source Registration with Response Body containing the key '["information"] [0]["entities"][0] ["type"]', with the value 'https://ngsi-ld-test-suite/context#Vehicle' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_09
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5
Config Id	CF_05
PICS Selection	PICS_HTTP_CONTEXT_RESOLV

Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_09_01	
Name	033_09_01 Create one context source registration with a Link header and a JSON-LD content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 400 and Create Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Context Source Registration With Return with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_10	
Test objective	Check that one cannot create a context source with invalid content	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_10_01	
Name	033_10_01 Create a context source registration with a different data structure than CSourceRegistration data type	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source with invalid content	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 400 and Create Context Source Registration With Return with Response Header: Location containing \$\$\${registration_id} } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/CSR/033_10_02	
Name	033_10_02 Create a context source registration with a date in the past	
Parent Release	v1.3.1	

Description	Check that one cannot create a context source with invalid content	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 400 and Create Context Source Registration With Return with Response Header: Location containing \${registration_id} } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/CSR/033_08	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/CSR/033_08_01	
Name	033_08_01 Create one context source registration not containing a JSON-LD @context with a JSON-LD content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: '\${updated_payload}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration With Return with Response Status Code set to 400 and Create Context Source Registration With Return with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Context Source Registration With Return with Response body containing 'title' element } </pre>	SUT -> Client

4.3.2 RegistrationSubscription

4.3.2.1 RetrieveContextSourceRegistrationSubscription

TP Id	TP/NGSI-LD/CS/REGSUB/040_01	
Test objective	Check that one can retrieve a context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubRet	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/040_01_01	
Name	040_01_01 Retrieve Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one can retrieve a context source registration subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'GET' Retrieve Context Source Registration Subscription with data: and Query Parameter: subscription id set to '\$ {subscription_id}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration Subscription with Response Status Code set to 200 and Retrieve Context Source Registration Subscription with Response reason set to 'OK' and Retrieve Context Source Registration Subscription with The response headers Link is set to 'expected_link_header=\$ {expected link header}' and Retrieve Context Source Registration Subscription with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'expectation_payload' and Query Parameter: 'response_body' set to 'response.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' list of keys and Retrieve Context Source Registration Subscription with The Response Body `\$ {response.json()}' might contain the key 'lastNotification' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body `\$ {response.json()}' might contain the key 'lastFailure' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body `\$ {response.json()}' might contain the key 'lastSuccess' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body `\$ {response.json()}' might contain the key 'timesSent' in the NotificationParams } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/040_03	
Test objective	Check that one cannot retrieve an unknown context source registration subscription, an error of type ResourceNotFound shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/040_03_01	
Name	040_03_01 Retrieve Unknown Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve an unknown context source registration subscription, an error of type ResourceNotFound shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'GET' Retrieve Context Source Registration Subscription with data: and Query Parameter: subscription id set to 'urn:ngsi-ld:Subscription:unknowSubscription' } then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration Subscription with Response Status Code set to 404 and Retrieve Context Source Registration Subscription with Response reason set to 'Not Found' and Retrieve Context Source Registration Subscription with Response Header: Content- Type set to application/ json and Retrieve Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

TP Id	TP/NGSI-LD/CS/REGSUB/040_02	
Test objective	Check that one cannot retrieve a context source registration subscription with an invalid URI, an error of type BadRequestData shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.4	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/040_02_01	
Name	040_02_01 Retrieve Context Source Registration Subscription With An Invalid Id	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a context source registration subscription with an invalid URI, an error of type BadRequestData shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'GET' Retrieve Context Source Registration Subscription with data: and Query Parameter: subscription id set to 'invalidUri' } </pre>	<p>SUT <- Client</p>

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration Subscription with Response Status Code set to 400 and Retrieve Context Source Registration Subscription with Response reason set to 'Bad Request' and Retrieve Context Source Registration Subscription with Response Header: Content- Type set to application/ json and Retrieve Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
--	---	---------------

4.3.2.2 ContextSourceRegistrationSubscriptionNotificationBehaviour

TP Id	TP/NGSI-LD/CS/REGSUB/047_16	
Test objective	Check if one updates a context source registration subscription, a CsourceNotification will be sent with all currently matching context source registrations	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a list of Context Source Registrations (CSRs) providing latest information about some entities and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_16_01	
Name	047_16_01 MatchFirstContextSourceRegistration	
Parent Release	v1.3.1	
Description	Check if one updates a context source registration subscription, a CsourceNotification will be sent with all currently matching context source registrations	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to '\$ {first_context_source_registration_id}' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_16_02	
Name	047_16_02 MatchSecondContextSourceRegistration	
Parent Release	v1.3.1	
Description	Check if one updates a context source registration subscription, a CsourceNotification will be sent with all currently matching context source registrations	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$' {subscription_id}' and subscription update fragment set to '\$' {subscription_update_fragment}' }	SUT <- Client
	then { the client at '\$' {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to '\$' {second_context_source_registration_id}' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$' {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_16_03	
Name	047_16_03 MatchBothContextSourceRegistrations	
Parent Release	v1.3.1	
Description	Check if one updates a context source registration subscription, a CsourceNotification will be sent with all currently matching context source registrations	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$' {subscription_id}' and subscription update fragment set to '\$' {subscription_update_fragment}' }	SUT <- Client
	then { the client at '\$' {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to '\$' {first_context_source_registration_id}' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$' {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_14
Test objective	Check if a context source registrations subscription defines a geoQ member, a CsourceNotification will be triggered from matching context source registrations with a matching location member
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7
Config Id	CF_05
PICS Selection	PICS_NOTIFICATIONS_CSR
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }
Permutations	
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_14_01
Name	047_14_01 Receive cSourceNotification For Matching Context Source Registrations On Location
Parent Release	v1.3.1

Description	Check if a context source registrations subscription defines a geoQ member, a CsourceNotification will be triggered from matching context source registrations with a matching location member	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content- Type set to 'application/ Id+json' and payload set to '\$ {context_source_registration_payload}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_10	
Test objective	Check if a context source registration subscription defines temporalQ member with timeproperty observedAt, the temporal query is matched against the observationInterval of matching context source registrations	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_10_01	
Name	047_10_01 Receive cSourceNotification For Matching Context Source Registrations On Observation Interval	
Parent Release	v1.3.1	
Description	Check if a context source registration subscription defines temporalQ member with timeproperty observedAt, the temporal query is matched against the observationInterval of matching context source registrations	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content- Type set to 'application/ Id+json' and payload set to '\$ {context_source_registration_payload}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_09	
Test objective	Check if a context source registration subscription defines an "entities" member, a CsourceNotification will be triggered from context source registrations with information member matching the described "entities"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_09_01	
Name	047_09_01 Receive cSourceNotification For No Longer Matching Context Source Registrations Providing Latest Information	
Parent Release	v1.3.1	
Description	Check if a context source registration subscription defines an "entities" member, a CsourceNotification will be triggered from context source registrations with information member matching the described "entities"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'PATCH' Update Context Source Registration with CSR Id set to '\$ {context_source_registration_id}' and subscription update fragment set to '\$ {update_fragment}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'noLongerMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_12	
Test objective	Check if a context source registrations subscription defines entities member and watchedAttributes member, a CsourceNotification will be triggered from context source registrations with information member matching the described "entities" and "attributes"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_12_01	
Name	047_12_01 Receive cSourceNotification For Matching Context Source Registrations On Watched Attributes	
Parent Release	v1.3.1	
Description	Check if a context source registrations subscription defines entities member and watchedAttributes member, a CsourceNotification will be triggered from context source registrations with information member matching the described "entities" and "attributes"	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content-Type set to 'application/ld+json' and payload set to '\$ {context_source_registration_payload}' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_15	
Test objective	Check if a context source registrations subscription does not define a geoproperty in the geoQ member, a CsourceNotification will be triggered from matching context source registrations with a matching location member	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_15_01	
Name	047_15_01 Receive cSourceNotification For Matching Context Source Registrations On Location As Default	
Parent Release	v1.3.1	
Description	Check if a context source registrations subscription does not define a geoproperty in the geoQ member, a CsourceNotification will be triggered from matching context source registrations with a matching location member	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content-Type set to 'application/ld+json' and payload set to '\$ {context_source_registration_payload}' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_13	
Test objective	Check if a context source registrations subscription does not define watchedAttributes member, a CsourceNotification will be triggered from context source registrations with information member matching all attributes of the described entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_13_01	
Name	047_13_01 Receive cSourceNotification For Matching Context Source Registrations On Any watchedAttribute	
Parent Release	v1.3.1	
Description	Check if a context source registrations subscription does not define watchedAttributes member, a CsourceNotification will be triggered from context source registrations with information member matching all attributes of the described entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content-Type set to 'application/ld+json' and payload set to '\$ {context_source_registration_payload}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_11	
Test objective	Check if a context source registration subscription defines temporalQ member with timeproperty createdAt or modifiedAt, the temporal query is matched against the managementInterval of matching context source registrations	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_11_01	
Name	047_11_01 CreatedAt	
Parent Release	v1.3.1	
Description	Check if a context source registration subscription defines temporalQ member with timeproperty createdAt or modifiedAt, the temporal query is matched against the managementInterval of matching context source registrations	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ </pre>	SUT <- Client

	Id+json' }	
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_11_02	
Name	047_11_02 ModifiedAt	
Parent Release	v1.3.1	
Description	Check if a context source registration subscription defines temporalQ member with timeproperty createdAt or modifiedAt, the temporal query is matched against the managementInterval of matching context source registrations	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ Id+json' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_08
Test objective	Check if a context source registration subscription does not define a temporalQ member, a CsourceNotification will be triggered from matching context source registrations for context sources providing latest information
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7
Config Id	CF_05
PICS Selection	PICS_NOTIFICATIONS_CSR
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration (CSR1) providing latest information about some entities and the SUT containing a Context Source Registration Subscription (CSRS1) }
Permutations	
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_08_01
Name	047_08_01 Receive cSourceNotification For Matching Context Source Registrations Providing Latest Information
Parent Release	v1.3.1
Description	Check if a context source registration subscription does not define a temporalQ member, a CsourceNotification will be triggered from matching context source registrations for context sources providing latest information

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'PATCH' Update Context Source Registration with CSR Id set to '\$ {context_source_registration_id}' and subscription update fragment set to '\$ {update_fragment}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'updated' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_04	
Test objective	The structure of the csource notification message shall be as mandated by clause 5.3.2	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_04_01	
Name	047_04_01 Receive cSourceNotification With Compliant Structure	
Parent Release	v1.3.1	
Description	The structure of the csource notification message shall be as mandated by clause 5.3.2	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content- Type set to 'application/ Id+json' and payload set to '\$ {context_source_registration_payload}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_06	
Test objective	Check that if a cSourceNotification is not sent successfully, the "notification.timesSent" member shall be incremented by one and the notification.lastFailure" and "notification.lastNotification" members shall be updated with the current timestamp and the status of the context source registration subscription shall be updated to "failed"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	

Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_06_01	
Name	047_06_01 If A cSourceNotification Is Not Successfully Sent The Notification Member Shall Be Updated	
Parent Release	v1.3.1	
Description	Check that if a cSourceNotification is not sent successfully, the "notification.timesSent" member shall be incremented by one and the notification.lastFailure" and "notification.lastNotification" members shall be updated with the current timestamp and the status of the context source registration subscription shall be updated to "failed"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'GET' Retrieve Context Source Registration Subscription with data: and Query Parameter: subscription id set to '\$ {subscription_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Retrieve Context Source Registration Subscription with Response containing: * Notification expectation file path set to 'notifications/ expectations/1-timesSent-failed.json' * Expected Additional Members set to '\$ {expected_notification_additional_members}' * Response Body set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_02	
Test objective	Check that if the created context source registration subscription does not define a timeInterval member, a cSourceNotification, with the appropriate trigger reason in the "triggerReason" member, will be sent initially on subscription and whenever there is a change of a matching Context Source Registration	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration (CSR1) providing latest information about some entities and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_02_01	
Name	047_02_01 Receive cSourceNotification Initially On Subscription And Whenever There Is A Change Of A Matching Context Source Registration	
Parent Release	v1.3.1	
Description	Check that if the created context source registration subscription does not define a timeInterval member, a cSourceNotification, with the appropriate trigger reason in the "triggerReason" member, will be sent initially on subscription and whenever there is a change of a matching Context Source Registration	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'PATCH' Update Context Source Registration with CSR Id set to '\$ {context_source_registration_id}' and subscription update fragment set to '\$ {update_fragment}' } </pre>	SUT <- Client

	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'updated' and Query Parameter: expected_notification_data_entities set to '\$ {EMPTY}' and Query Parameter: timeout set to '5' } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CS/REGSUB/047_07	
Test objective	Check that a cSourceNotification shall only be sent if and only if the status of the corresponding subscription is active, neither paused or expired	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_07_01	
Name	047_07_01 PausedSubscription	
Parent Release	v1.3.1	
Description	Check that a cSourceNotification shall only be sent if and only if the status of the corresponding subscription is active, neither paused or expired	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for no Notification data and Query Parameter: timeout set to '5' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_07_02	
Name	047_07_02 ExpiredSubscription	
Parent Release	v1.3.1	
Description	Check that a cSourceNotification shall only be sent if and only if the status of the corresponding subscription is active, neither paused or expired	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for no Notification data and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_05	
Test objective	Check that if a cSourceNotification is sent successfully to the "endpoint" member, the "notification.timesSent" member shall be incremented by one and the "notification.lastSuccess" and "notification.lastNotification" members shall be updated with the current timestamp and the status of the context source registration subscription shall be updated to "ok"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_05_01	
Name	047_05_01 If A cSourceNotification Is Successfully Sent The Notification Member Shall Be Updated	
Parent Release	v1.3.1	
Description	Check that if a cSourceNotification is sent successfully to the "endpoint" member, the "notification.timesSent" member shall be incremented by one and the "notification.lastSuccess" and "notification.lastNotification" members shall be updated with the current timestamp and the status of the context source registration subscription shall be updated to "ok"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'GET' Retrieve Context Source Registration Subscription with data: and Query Parameter: subscription id set to '\$ {subscription_id}' } then { the SUT sends a valid Response for the operation: Retrieve Context Source Registration Subscription with Response containing: * Notification expectation file path set to 'notifications/ expectations/1-timesSent-ok.json' * Expected Additional Members set to '\$ {expected_notification_additional_members}' * Response Body set to '\$ {response.json()}' } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_01	
Test objective	Check that if the created context source registration subscription defines a timeInterval member, a cSourceNotification will be sent periodically, initially on subscription and when the time interval is reached	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_01_01	
Name	047_01_01 Receive cSourceNotification Periodically And Initially On Subscription	
Parent Release	v1.3.1	
Description	Check that if the created context source registration subscription defines a timeInterval member, a cSourceNotification will be sent periodically, initially on subscription and when the time interval is reached	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '15' seconds }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/047_03	
Test objective	Check that instead of providing the original context source registration, implementations should return context source registration information relevant for the subscription, in particular only matching RegistrationInfo elements	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.7	
Config Id	CF_05	
PICS Selection	PICS_NOTIFICATIONS_CSR	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/047_03_01	
Name	047_03_01 Receive cSourceNotification With Relevant Information	
Parent Release	v1.3.1	
Description	Check that instead of providing the original context source registration, implementations should return context source registration information relevant for the subscription, in particular only matching RegistrationInfo elements	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'POST' Create Context Source Registration Request with Content- Type set to 'application/ ld+json' and payload set to '\$ {context_source_registration_payload}' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for Notification and validate it and Query Parameter: expected_subscription_id set to 'subscription_id' and Query Parameter: expected_context_source_registration_ids set to 'expected_context_source_registration_ids' and Query Parameter: expected_trigger_reason set to 'newlyMatching' and Query Parameter: expected_notification_data_entities set to 'expected_notification_data_entities' and Query Parameter: timeout set to '5' }	SUT -> Client

4.3.2.3 DeleteContextSourceRegistrationSubscription

TP Id	TP/NGSI-LD/CS/REGSUB/042_03	
Test objective	Check that one cannot delete an unknown context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.6	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubDel	
Initial conditions	with { the SUT containing an initial state }	

Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/042_03_01	
Name	042_03_01 Delete Unknown Context Source Registration Subscription With Invalid Uri	
Parent Release	v1.3.1	
Description	Check that one cannot delete an unknown context source registration subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'DELETE' Delete Context Source Registration Subscription with id set to 'urn:ngsi-ld:Subscription:unknowSubscription' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Context Source Registration Subscription with Response Status Code set to 404 and Delete Context Source Registration Subscription with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/ResourceNotFound' and Delete Context Source Registration Subscription with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/042_01	
Test objective	Check that one can delete a context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.6	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/042_01_01	
Name	042_01_01 Delete Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one can delete a context source registration subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'DELETE' Delete Context Source Registration Subscription with id set to '\$ {subscription_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Context Source Registration Subscription with Response Status Code set to 204 and Retrieve Context Source Registration Subscription with Response Status Code set to \${response1.status_code} } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/042_02
Test objective	Check that one cannot delete a context source registration subscription with an invalid URI
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.6
Config Id	CF_05
PICS Selection	PICS_CsoRegSubDel
Initial conditions	with { the SUT containing an initial state }

Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/042_02_01	
Name	042_02_01 Delete Context Source Registration Subscription With Invalid Uri	
Parent Release	v1.3.1	
Description	Check that one cannot delete a context source registration subscription with an invalid URI	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'DELETE' Delete Context Source Registration Subscription with id set to 'invalidUri' } then { the SUT sends a valid Response for the operations: Delete Context Source Registration Subscription with Response Status Code set to 400 and Delete Context Source Registration Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Delete Context Source Registration Subscription with Response body containing 'title' element } </pre>	SUT <- Client
		SUT -> Client

4.3.2.4 CreateContextSourceRegistrationSubscription

TP Id	TP/NGSI-LD/CS/REGSUB/038_02	
Test objective	Check that one can create a context source registration subscription without providing an id and it will be automatically generated	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_02_01	
Name	038_02_01 Create Context Source Registration Subscription Without An Id	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration subscription without providing an id and it will be automatically generated	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Retrieve Context Source Registration Subscription with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/038_04	
Test objective	Check that one can create a context source registration subscription with isActive member set to false and its initial status will be set to "paused"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_04_01	
Name	038_04_01 Create Inactive Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration subscription with isActive member set to false and its initial status will be set to "paused"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Create Context Source Registration Subscription with Response Header: Location containing \$\${subscription_id} and Retrieve Context Source Registration Subscription with Check that the payload body contains a boolean attribute with name 'expected_attribute_name=isActive' and value 'expected_attribute_value=False' and Retrieve Context Source Registration Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'paused' } </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

TP Id	TP/NGSI-LD/CS/REGSUB/038_06	
Test objective	Check that one can create a context source registration subscription without an expiresAt member and it will be considered as perpetual	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_06_01	
Name	038_06_01 Create Context Source Registration Subscription Without expiresAt Member	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration subscription without an expiresAt member and it will be considered as perpetual	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ </pre>	<p>SUT <- Client</p>

	<pre> Id+json' } then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Create Context Source Registration Subscription with Response Header: Location containing \$\$subscription_id} and Retrieve Context Source Registration Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'active' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CS/REGSUB/038_01	
Test objective	Check that one can create a minimal context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_01_01	
Name	038_01_01 Create Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one can create a minimal context source registration subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Create Context Source Registration Subscription with Response Header: Location containing \$\$\${subscription_id} and Retrieve Context Source Registration Subscription with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys and Retrieve Context Source Registration Subscription with The Response Body '\$ {response1.json()}' might contain the key 'lastNotification' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body '\$ {response1.json()}' might contain the key 'lastNotification' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body '\$ {response1.json()}' might contain the key 'lastNotification' in the NotificationParams and Retrieve Context Source Registration Subscription with The Response Body '\$ {response1.json()}' might contain the key 'timesSent' in the NotificationParams } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CS/REGSUB/038_03	
Test objective	Check that one can create a context source registration subscription without providing isActive member and will be active by default	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_03_01	
Name	038_03_01 Create Context Source Registration Subscription Without isActive Member	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration subscription without providing isActive member and will be active by default	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Create Context Source Registration Subscription with Response Header: Location containing \$\$\${subscription_id} and Retrieve Context Source Registration Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and </pre>	SUT -> Client

	Query Parameter: expected_attribute_value set to 'active'	
--	---	--

TP Id	TP/NGSI-LD/CS/REGSUB/038_07	
Test objective	Check that one cannot create a context source registration subscription where another context source registration subscription whose id is equivalent exists	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a Context Source Registration Subscription (CSRS1) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_07_01	
Name	038_07_01 Create Existing Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration subscription where another context source registration subscription whose id is equivalent exists	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 409 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_ALREADY_EXISTS}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

TP Id	TP/NGSI-LD/CS/REGSUB/038_05	
Test objective	Check that one can create a context source registration subscription with an expiresAt member and when it is due the status of the subscription changes to "expired"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_05_01	
Name	038_05_01 Create Context Source Registration Subscription With expiresAt Member	
Parent Release	v1.3.1	
Description	Check that one can create a context source registration subscription with an expiresAt member and when it is due the status of the subscription changes to "expired"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ </pre>	<p>SUT <- Client</p>

	<pre> Id+json' } then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 201 and Create Context Source Registration Subscription with Response Header: Location containing \${subscription_id} and Retrieve Context Source Registration Subscription with Check that the dictionary contains the key 'key=expiresAt' with type DateTime and value set to 'expected value=\${ {expiresAt}' and Retrieve Context Source Registration Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'expired' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CS/REGSUB/038_09	
Test objective	Check that one cannot create a context source registration subscription with an expiration timestamp representing a moment before the current date and time	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	
PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_09_01	
Name	038_09_01 Create Expired Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration subscription with an expiration timestamp representing a moment before the current date and time	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 400 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\${ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/038_08	
Test objective	Check that one cannot create a context source registration subscription If the data types, cardinalities and restrictions expressed by clause 5.2.12 are not met	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.2	
Config Id	CF_05	

PICS Selection	PICS_CreSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_08_01	
Name	038_08_01 WithoutNotification	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration subscription If the data types, cardinalities and restrictions expressed by clause 5.2.12 are not met	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 400 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_08_02	
Name	038_08_02 InvalidType	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration subscription If the data types, cardinalities and restrictions expressed by clause 5.2.12 are not met	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 400 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_08_03	
Name	038_08_03 InvalidQuery	
Parent Release	v1.3.1	

Description	Check that one cannot create a context source registration subscription If the data types, cardinalities and restrictions expressed by clause 5.2.12 are not met	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 400 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/038_08_04	
Name	038_08_04 EmptyWatchedAttributes	
Parent Release	v1.3.1	
Description	Check that one cannot create a context source registration subscription If the data types, cardinalities and restrictions expressed by clause 5.2.12 are not met	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/csourceSubscriptions' method set to 'POST' Create Context Source Registration Subscription with filename set to '\$ {subscription_payload}', accept set to '\$ {EMPTY}', and content-type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Context Source Registration Subscription with Response Status Code set to 400 and Create Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

4.3.2.5 UpdateContextSourceRegistrationSubscription

TP Id	TP/NGSI-LD/CS/REGSUB/039_02
Test objective	Check that one cannot update a context source registration subscription with an invalid URI
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.3
Config Id	CF_05
PICS Selection	PICS_CsoRegSubUpd
Initial conditions	with { the SUT containing an initial state }

Permutations							
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_02_01						
Name	039_02_01 Update Context Source Registration Subscription With Invalid Uri						
Parent Release	v1.3.1						
Description	Check that one cannot update a context source registration subscription with an invalid URI						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to 'invalidUri' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </td> <td>SUT <- Client</td> </tr> <tr> <td></td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to 'invalidUri' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT <- Client		SUT -> Client
	Test events	Direction					
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to 'invalidUri' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT <- Client						
	SUT -> Client						

TP Id	TP/NGSI-LD/CS/REGSUB/039_04						
Test objective	Check that one cannot update a context source registration subscription with a fragment that does not meet the data types and restrictions expressed by clause 5.2.12						
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.3						
Config Id	CF_05						
PICS Selection	PICS_CsoRegSubUpd						
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSs) }						
Permutations							
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_04_01						
Name	039_04_01 InvalidType						
Parent Release	v1.3.1						
Description	Check that one cannot update a context source registration subscription with a fragment that does not meet the data types and restrictions expressed by clause 5.2.12						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not </td> <td>SUT <- Client</td> </tr> <tr> <td></td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not	SUT <- Client		SUT -> Client
	Test events	Direction					
when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not	SUT <- Client						
	SUT -> Client						

	<pre> \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_04_02	
Name	039_04_02 InvalidNotification	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration subscription with a fragment that does not meet the data types and restrictions expressed by clause 5.2.12	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 400 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/039_01	
Test objective	Check that one can update a context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSs) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_01_01	
Name	039_01_01 Update Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one can update a context source registration subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to '\$ {subscription_id}' and subscription update fragment set to '\$ {subscription_update_fragment}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 204 and Update Context Source Registration Subscription with Response does not contain a body and Update Context Source Registration Subscription with Response reason set to 'No Content' and </pre>	SUT -> Client

	Retrieve Context Source Registration Subscription with Check Updated Entity and Query Parameter: 'updated_resource' set to 'subscription' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' }	
--	--	--

TP Id	TP/NGSI-LD/CS/REGSUB/039_03	
Test objective	Check that one cannot update an unknown context source registration subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_03_01	
Name	039_03_01 Update Unknown Context Source Registration Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot update an unknown context source registration subscription	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}' method set to 'PATCH' Update Context Source Registration Subscription with subscription id set to 'urn:ngsi-ld:Subscription:unknowSubscription' and subscription update fragment set to '\$ {subscription_update_fragment}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription with Response Status Code set to 404 and Update Context Source Registration Subscription with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_RESOURCE_NOT_FOUND}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/039_05	
Test objective	Check that one cannot update a context source registration subscription with an invalid request body (invalid JSON document)	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.3	
Config Id	CF_05	
PICS Selection	PICS_CsoRegSubUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSSs) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/039_05_01	
Name	039_05_01 Update Context Source Registration Subscription With Invalid JSON Fragment	
Parent Release	v1.3.1	
Description	Check that one cannot update a context source registration subscription with an invalid request body (invalid JSON document)	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions/{subscription_id}'	SUT <- Client

	<pre> method set to 'PATCH' Update Context Source Registration Subscription from file with subscription id set to '\$ {subscription_id}' and subscription update fragment set to 'csourceSubscriptions/ fragments/subscription-update-invalid-json-sample.json' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Update Context Source Registration Subscription From File with Response Status Code set to 400 and Update Context Source Registration Subscription From File with Response Body containing the type 'problem_type=\$ {ERROR_TYPE_BAD_REQUEST_DATA}' and Response Body Title is a string and is not \${EMPTY} and Response Body Detail is a string and is not \${EMPTY} and Response Boty Title is not equal to Response Body Detail } </pre>	SUT -> Client

4.3.2.6 QueryContextSourceRegistrationSubscriptions

TP Id	TP/NGSI-LD/CS/REGSUB/041_01	
Test objective	Check that one can query context source registration subscriptions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.5	
Config Id	CF_05	
PICS Selection	PICS_RetLisSubCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSs) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_01_01	
Name	041_01_01 Query Context Source Registration Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response containing: * file set to 'csourceSubscriptions/ expectations/subscriptions-035-01-expectation.json' * id set to '\$ {subscription_ids}' * response set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/041_03	
Test objective	Check that one can query context source registration subscriptions with providing page and limit parameters for pagination	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.5	
Config Id	CF_05	

PICS Selection	PICS_RetLisSubCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSs) }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_03_01	
Name	041_03_01 Query Second Subscription	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with providing page and limit parameters for pagination	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${1}' and Query Parameter: page set to '\${2}' , } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${1}' ' of type 'Subscription' with response set to '\$ {response.json()}' and Query Context Source Registration Subscriptions with Response header with 'Prev' header equal to '</ ngsi-ld/v1/csourceSubscriptions? limit=1&page=3>;rel="next";type="application/ ld+json"' and with 'Next' header equal to '\$ {response.headers}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_03_02	
Name	041_03_02 Query Last Subscription	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with providing page and limit parameters for pagination	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${2}' and Query Parameter: page set to '\${2}' , } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${1}' ' of type 'Subscription' with response set to '\$ </pre>	SUT -> Client

	{response.json()}' and Query Context Source Registration Subscriptions with Response header with 'Prev' header equal to '\$ {EMPTY}' and with 'Next' header equal to '\$ {response.headers}' }	
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_03_03	
Name	041_03_03 Query All Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with providing page and limit parameters for pagination	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${15}' ' and Query Parameter: page set to '\${1}' , }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${3}' ' of type 'Subscription' with response set to '\$ {response.json()}' and Query Context Source Registration Subscriptions with Response header with 'Prev' header equal to '\$ {EMPTY}' and with 'Next' header equal to '\$ {response.headers}' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/041_02	
Test objective	Check that one can query context source registration subscriptions with a limit parameter and it will be the maximum number of subscriptions to be retrieved	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.5	
Config Id	CF_05	
PICS Selection	PICS_RetLisSubCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a set of Context Source Registration Subscriptions (CSRSs) }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_02_01	
Name	041_02_01 Query One Subscription	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with a limit parameter and it will be the maximum number of subscriptions to be retrieved	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${1}' }	SUT <- Client

	<pre> } then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${1}' ' of type 'Subscription' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_02_02	
Name	041_02_02 Query Two Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with a limit parameter and it will be the maximum number of subscriptions to be retrieved	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${2}' , } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${2}' ' of type 'Subscription' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_02_03	
Name	041_02_03 Query All Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query context source registration subscriptions with a limit parameter and it will be the maximum number of subscriptions to be retrieved	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${15}' , } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 200 and Query Context Source Registration Subscriptions with Response Body containing a list of entities equal to '\${3}' ' of type 'Subscription' with response set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/REGSUB/041_04	
Test objective	Check that one cannot query context source registration subscriptions with invalid page and limit parameters	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.11.5	
Config Id	CF_05	
PICS Selection	PICS_RetLisSubCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_04_01	
Name	041_04_01 Invalid Limit	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registration subscriptions with invalid page and limit parameters	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: limit set to '\${-5}' and Query Parameter: page set to '\${2}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 400 and Query Context Source Registration Subscriptions with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Context Source Registration Subscriptions with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_04_02	
Name	041_04_02 Invalid Page	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registration subscriptions with invalid page and limit parameters	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: limit set to '\${2}' and Query Parameter: page set to '\${-3}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 400 and Query Context Source Registration Subscriptions with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Context Source Registration Subscriptions with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/REGSUB/041_04_03	
Name	041_04_03 Invalid Limit And Page	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registration subscriptions with invalid page and limit parameters	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/csourceSubscriptions' method set to 'GET' Query Context Source Registration Subscriptions and Query Parameter: limit set to '\${0}' and Query Parameter: page set to '\${0}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registration Subscriptions with Response Status Code set to 400 and Query Context Source Registration Subscriptions with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Context Source Registration Subscriptions with Response body containing 'title' element }	SUT -> Client

4.3.3 Discovery

4.3.3.1 RetrieveContextSourceRegistration

TP Id	TP/NGSI-LD/CS/DISC/036_01	
Test objective	Check that one cannot retrieve a Context Source Registration, if the context source registration id is not a valid URI	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.1	
Config Id	CF_05	
PICS Selection	PICS_CsoRegRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/036_01_01	
Name	036_01_01 Invalid Id	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a Context Source Registration, if the context source registration id is not a valid URI	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-lid/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to 'invalidUri' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 400 and Retrieve Context Source Registration with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Retrieve Context Source Registration with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/036_03
Test objective	Check that one can retrieve a Context Source Registration. Term to URI expansion of Attribute names shall be observed.

Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.1	
Config Id	CF_05	
PICS Selection	PICS_CsoRegRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/036_03_01	
Name	036_03_01 Retrieve Context Source Registration With Default Core Context	
Parent Release	v1.3.1	
Description	Check that one can retrieve a Context Source Registration. Term to URI expansion of Attribute names shall be observed.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to '\$ {context_source_registration_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 200 and Retrieve Context Source Registration with Response containing: * file set to 'csourceRegistrations/ expectations/context-source-registration-expanded-format.json' * id set to '\$ {context_source_registration_id}' * response set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/036_05	
Test objective	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_05	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/036_05_01	
Name	036_05_01 EmptyJsonLdContext	
Parent Release	v1.3.1	
Description	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to '\$ {context_source_registration_id}' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 200 and </pre>	SUT -> Client

	Retrieve Context Source Registration with Response containing: * file set to 'csourceRegistrations/expectations/context-source-registration-expanded-format.json' * id set to '\$ {context_source_registration_id}' * response set to '\$ {response.json()}' }	
TP Permutation id	TP/NGSI-LD/CS/DISC/036_05_02	
Name	036_05_02 CreationTimeJsonLdContext	
Parent Release	v1.3.1	
Description	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to '\$ {context_source_registration_id}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 200 and Retrieve Context Source Registration with Response containing: * file set to 'csourceRegistrations/expectations/context-source-registration.json' * id set to '\$ {context_source_registration_id}' * response set to '\$ {response.json()}' }	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/036_02	
Test objective	Check that one cannot retrieve a Context Source Registration, if the NGSI-LD endpoint does not know about the target context source registration, because there is no existing context source registration whose id (URI) is equivalent	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.1	
Config Id	CF_05	
PICS Selection	PICS_CsoRegRet	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CS/DISC/036_02_01	
Name	036_02_01 Retrieve Unknown Context Source Registration	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a Context Source Registration, if the NGSI-LD endpoint does not know about the target context source registration, because there is no existing context source registration whose id (URI) is equivalent	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to 'urn:ngsi-ld:ContextSourceRegistration:unknowRegistration' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 404 and Retrieve Context Source Registration with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Context Source Registration with Response body containing 'title' element } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CS/DISC/036_04	
Test objective	Check that one can retrieve a Context Source Registration	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.1	
Config Id	CF_05	
PICS Selection	PICS_CsoRegRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/036_04_01	
Name	036_04_01 Retrieve Context Source Registration	
Parent Release	v1.3.1	
Description	Check that one can retrieve a Context Source Registration	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations/{context_source_registration_id}' method set to 'GET' Retrieve Context Source Registration with data: and Query Parameter: context source registration id set to '\$ {context_source_registration_id}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Context Source Registration with Response Status Code set to 200 and Retrieve Context Source Registration with Response containing: * file set to 'csourceRegistrations/ expectations/context-source-registration.json' * id set to '\$ {context_source_registration_id}' * response set to '\$ {response.json()}' } </pre>	SUT -> Client

4.3.3.2 QueryContextSourceRegistrations

TP Id	TP/NGSI-LD/CS/DISC/037_08	
Test objective	Check that one can query context source registrations. If no temporal query is present, only Context Source Registrations for Context Sources providing latest information are considered	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_08_01	
Name	037_08_01 Query Context Source Registration Without Temporal Query	

Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If no temporal query is present, only Context Source Registrations for Context Sources providing latest information are considered	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Vehicle' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_11	
Test objective	Check that one can query context source registrations with providing page and limit parameters, pagination logic shall be in place as mandated by clause 5.5.9.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a list of Context Source Registrations (CSRs) providing latest information about some entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_11_01	
Name	037_11_01 Query Second Subscription	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations with providing page and limit parameters, pagination logic shall be in place as mandated by clause 5.5.9.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: limit set to '\${1}' ' and Query Parameter: offset set to '\${2}' ' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Query Context Source Registrations with Response Body containing a list of entities equal to '\${1}' ' of type 'ContextSourceRegistration' with response set to '\$ {response.json()}' and Query Context Source Registrations with Response header with 'Prev' header equal to '</ ngsi-ld/v1/csourceSubscriptions? type=Building&limit=1&offset=3>;rel="next";type="application/ </pre>	SUT -> Client

	ld+json'" and with 'Next' header equal to '\$ {response.json()}' }	
TP Permutation id	TP/NGSI-LD/CS/DISC/037_11_02	
Name	037_11_02 Query Last Subscription	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations with providing page and limit parameters, pagination logic shall be in place as mandated by clause 5.5.9.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: limit set to '\${2}' ' and Query Parameter: offset set to '\${2}' ' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Query Context Source Registrations with Response Body containing a list of entities equal to '\${1}' ' of type 'ContextSourceRegistration' with response set to '\$ {response.json()}' and Query Context Source Registrations with Response header with 'Prev' header equal to '\$ {EMPTY}' and with 'Next' header equal to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_11_03	
Name	037_11_03 Query All Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations with providing page and limit parameters, pagination logic shall be in place as mandated by clause 5.5.9.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: limit set to '\${15}' ' and Query Parameter: offset set to '\${0}' ' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Query Context Source Registrations with Response Body containing a list of entities equal to '\${2}' ' of type 'ContextSourceRegistration' with response set to '\$ {response.json()}' and Query Context Source Registrations with Response header with 'Prev' header equal to '\$ </pre>	SUT -> Client

	{EMPTY}' and with 'Next' header equal to '\$ {response.json()}' }	
--	---	--

TP Id	TP/NGSI-LD/CS/DISC/037_09	
Test objective	Check that one can query context source registrations. If present, the temporal query is matched against the observationInterval or the managementInterval	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_09_01	
Name	037_09_01 Observation Interval With observedAt	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the temporal query is matched against the observationInterval or the managementInterval	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: timeproperty set to 'observedAt' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2021-08-01T22:00:00Z' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_09_02	
Name	037_09_02 Observation Interval Without timeproperty	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the temporal query is matched against the observationInterval or the managementInterval	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: timeproperty set to '\$ {EMPTY}' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2021-08-01T22:00:00Z' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context </pre>	SUT -> Client

	Source Registrations resolved against the default JSON-LD context }	
TP Permutation id	TP/NGSI-LD/CS/DISC/037_09_03	
Name	037_09_03 Management Interval With createdAt	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the temporal query is matched against the observationInterval or the managementInterval	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: timeproperty set to 'createdAt' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2021-08-01T22:00:00Z' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_09_04	
Name	037_09_04 Management Interval With modifiedAt	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the temporal query is matched against the observationInterval or the managementInterval	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: timeproperty set to 'modifiedAt' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2021-08-01T22:00:00Z' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_10
Test objective	Check that one can query context source registrations. If present, the conditions specified by the context source query match the respective Context Source Properties
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2
Config Id	CF_05
PICS Selection	PICS_DisCsoReg

Initial conditions	with { the SUT being in the "initial state" and the SUT containing a list of Context Source Registrations (CSRs) providing latest information about some entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_10_01	
Name	037_10_01 With list of Entity Ids	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the conditions specified by the context source query match the respective Context Source Properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: id set to '\$ {first_context_source_registration_id},\$ {third_context_source_registration_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_10_02	
Name	037_10_02 With NGSI-LD Query	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the conditions specified by the context source query match the respective Context Source Properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: q set to 'csourceProperty1=="aValue"' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_01
Test objective	Check that one can query context source registrations if at least one of list of Entity Types or list of Attribute names is present
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2
Config Id	CF_05
PICS Selection	PICS_DisCsoReg
Initial conditions	with { the SUT being in the "initial state" and the SUT containing a list of Context Source Registrations (CSRs) providing latest information about some entities }
Permutations	
TP Permutation id	TP/NGSI-LD/CS/DISC/037_01_01

Name	037_01_01 With list of entity types	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations if at least one of list of Entity Types or list of Attribute names is present	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_01_02	
Name	037_01_02 With list of attribute names	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations if at least one of list of Entity Types or list of Attribute names is present	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: attr set to 'name' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_03
Test objective	Check that one cannot query context source registrations, if the list of Entity identifiers includes a URI which it is not valid, or the query, geo-query or temporal query are not syntactically valid
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2
Config Id	CF_05
PICS Selection	PICS_DisCsoReg
Initial conditions	with { the SUT containing an initial state }
	Permutations
TP Permutation id	TP/NGSI-LD/CS/DISC/037_03_01
Name	037_03_01 Invalid URI
Parent Release	v1.3.1
Description	Check that one cannot query context source registrations, if the list of Entity identifiers includes a URI which it is not valid, or the query, geo-query or temporal query are not syntactically valid

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: id set to 'invalidUri' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 400 and Query Context Source Registrations with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Context Source Registrations with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_03_02	
Name	037_03_02 Invalid Query	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registrations, if the list of Entity identifiers includes a URI which it is not valid, or the query, geo-query or temporal query are not syntactically valid	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: q set to '(invalidQuery' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 400 and Query Context Source Registrations with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Context Source Registrations with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_03_03	
Name	037_03_03 Invalid GeoQuery	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registrations, if the list of Entity identifiers includes a URI which it is not valid, or the query, geo-query or temporal query are not syntactically valid	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and	SUT <- Client

	<pre> Query Parameter: georel set to 'within' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 400 and Query Context Source Registrations with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Context Source Registrations with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_03_04	
Name	037_03_04 Invalid Temporal Query	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registrations, if the list of Entity identifiers includes a URI which it is not valid, or the query, geo-query or temporal query are not syntactically valid	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-lid-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-lid-test-suite-compound.jsonld' and Query Parameter: timerel set to 'before' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 400 and Query Context Source Registrations with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Context Source Registrations with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_07	
Test objective	Check that one can query context source registrations. If present, the geoquery is matched against the GeoProperty programmatic parameter identified in the geoquery	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CS/DISC/037_07_01	
Name	037_07_01 Near Point	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the geoquery is matched against the GeoProperty programmatic parameter identified in the geoquery	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// </pre>	SUT <- Client

	forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: georel set to 'near;maxDistance==2000' and Query Parameter: geometry set to 'Point' and Query Parameter: coordinates set to '[-8.503,41.202] ' and Query Parameter: geoproperty set to '\$ {EMPTY}' }	
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_07_02	
Name	037_07_02 Within Polygon	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If present, the geoquery is matched against the GeoProperty programmatic parameter identified in the geoquery	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: georel set to 'within' and Query Parameter: geometry set to 'Polygon' and Query Parameter: coordinates set to '[[[-13.503,47.202], [6.541, 52.961],[20.37,44.653],[9.46,32.57], [-15.23,21.37]]' and Query Parameter: geoproperty set to 'location' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_05
Test objective	Check that one can query context source registrations matching EntityInfo of RegistrationInfo
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2
Config Id	CF_05
PICS Selection	PICS_DisCsoReg
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CS/DISC/037_05_01
Name	037_05_01 Registration With EntityInfo Matching The Query
Parent Release	v1.3.1
Description	Check that one can query context source registrations matching EntityInfo of RegistrationInfo

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'OffStreetParking' and Query Parameter: idPattern set to '.*downtown\$' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_05_02	
Name	037_05_02 Registration Without EntityInfo	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations matching EntityInfo of RegistrationInfo	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'OffStreetParking' and Query Parameter: idPattern set to '.*downtown\$' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_02	
Test objective	Check that one cannot query context source registrations, if neither Entity types nor Attribute names are provided, an error of type 400 shall be raised.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_02_01	
Name	037_02_01 Query Context Source Registrations Without Entity Types and Attribute Names	
Parent Release	v1.3.1	
Description	Check that one cannot query context source registrations, if neither Entity types nor Attribute names are provided, an error of type 400 shall be raised.	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations'	SUT <- Client

	<pre> method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 400 and Query Context Source Registrations with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Context Source Registrations with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_04	
Test objective	Check that one can query context source registrations. If a JSON-LD context is not provided, then all the query terms shall be resolved against the default JSON-LD @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_04_01	
Name	037_04_01 Query Context Source Registrations Without Context	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations. If a JSON-LD context is not provided, then all the query terms shall be resolved against the default JSON-LD @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\${context_source_registration_id}' and Query Parameter: type set to 'https://ngsi-ld-test-suite/context#Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CS/DISC/037_06	
Test objective	Check that one can query context source registrations matching property and relationship names of RegistrationInfo	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.10.2	
Config Id	CF_05	
PICS Selection	PICS_DisCsoReg	
Initial conditions	with { the SUT being in the "initial state" and the SUT contains a Context Source Registration with id equal to \${context_source_registration_id} and payload set to \${context_source_registration_payload_file_path} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CS/DISC/037_06_01	
Name	037_06_01 Query With Matching Properties And Relationships	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations matching property and relationship names of RegistrationInfo	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: attrs set to 'name,locatedAt' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CS/DISC/037_06_02	
Name	037_06_02 Query Without Properties And Relationships	
Parent Release	v1.3.1	
Description	Check that one can query context source registrations matching property and relationship names of RegistrationInfo	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/csourceRegistrations' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: type set to 'Building' and Query Parameter: attrs set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Context Source Registrations with Response Status Code set to 200 and Notification with Response body set to list of all matching Context Source Registrations resolved against the default JSON- LD context }	SUT -> Client

4.4 ContextInformation

4.4.1 Consumption

4.4.1.1 TemporalEntity

4.4.1.1.1 RetrieveTemporalEvolutionOfEntity

TP Id	TP/NGSI-LD/CI/Cons/TE/020_08
Test objective	Check that one cannot retrieve the temporal evolution of non-existing entity attributes
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3
Config Id	CF_01
PICS Selection	PICS_TemEvoEntRet
Initial conditions	with { the SUT containing an initial state }

Permutations							
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_08_01						
Name	020_08_01 Retrieve the temporal evolution of non-existing entity attributes						
Parent Release	v1.3.1						
Description	Check that one cannot retrieve the temporal evolution of non-existing entity attributes						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> <pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {temporal_attributes_to_be_retrieved}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre> </td> <td>SUT <- Client</td> </tr> <tr> <td> <pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 404 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element } </pre> </td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {temporal_attributes_to_be_retrieved}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 404 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element } </pre>	SUT -> Client
Test events	Direction						
<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {temporal_attributes_to_be_retrieved}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client						
<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 404 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element } </pre>	SUT -> Client						

TP Id	TP/NGSI-LD/CI/Cons/TE/020_11				
Test objective	Check that one can retrieve the temporal evolution of an entity with the aggregated temporal representation				
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.5.19, 5.7.3				
Config Id	CF_01				
PICS Selection	PICS_AGGRTEMPORAL and PICS_TemEvoEntRet				
Initial conditions	with { the SUT containing an initial state }				
Permutations					
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_11_01				
Name	020_11_01 One aggregate method aggregated by one hour duration				
Parent Release	v1.4.1				
Description	Check that one can retrieve the temporal evolution of an entity with the aggregated temporal representation				
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> <pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {EMPTY}' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'avg' and Query Parameter: aggrPeriodDuration set to 'PT1H' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre> </td> <td>SUT <- Client</td> </tr> </tbody> </table>	Test events	Direction	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {EMPTY}' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'avg' and Query Parameter: aggrPeriodDuration set to 'PT1H' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
Test events	Direction				
<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {EMPTY}' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'avg' and Query Parameter: aggrPeriodDuration set to 'PT1H' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client				

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-11-01-expectation.json' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_11_02	
Name	020_11_02 One aggregate method aggregated by one hour duration asking for one attribute	
Parent Release	v1.4.1	
Description	Check that one can retrieve the temporal evolution of an entity with the aggregated temporal representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to 'fuelLevel' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'avg' and Query Parameter: aggrPeriodDuration set to 'PT1H' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-11-02-expectation.json' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_11_03	
Name	020_11_03 Multiple aggregate methods aggregated by one hour duration	
Parent Release	v1.4.1	
Description	Check that one can retrieve the temporal evolution of an entity with the aggregated temporal representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ </pre>	SUT <- Client

	<pre>{temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {EMPTY}' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'avg,max' and Query Parameter: aggrPeriodDuration set to 'PT1H' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</pre>	
	<pre>then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-11-03-expectation.json' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' }</pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_11_04	
Name	020_11_04 Multiple aggregate methods aggregated by one day duration	
Parent Release	v1.4.1	
Description	Check that one can retrieve the temporal evolution of an entity with the aggregated temporal representation	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {EMPTY}' and Query Parameter: options set to '\$ {options}' and Query Parameter: aggrMethods set to 'min,max' and Query Parameter: aggrPeriodDuration set to 'P1D' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-11-04-expectation.json' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' }</pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_09	
Test objective	Check that one cannot retrieve the temporal evolution of an entity with an invalid request content	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${attrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_09_01	
Name	020_09_01 After	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve the temporal evolution of an entity with an invalid request content	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '\$ {EMPTY}' and Query Parameter: endTimeAt set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 400 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_09_02	
Name	020_09_02 Before	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve the temporal evolution of an entity with an invalid request content	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '\$ {EMPTY}' and Query Parameter: endTimeAt set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 400 and </pre>	SUT -> Client

	Retrieve Temporal Representation Of Entity with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element }	
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_09_03	
Name	020_09_03 Between	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve the temporal evolution of an entity with an invalid request content	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: timerel set to 'between' and Query Parameter: timeAt set to '2020-08-01T12:00:00Z' and Query Parameter: endTimeAt set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 400 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_10	
Test objective	Check that one can retrieve the temporal evolution of an entity with the simplified temporal representation	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_10_01	
Name	020_10_01 Retrieve the temporal evolution of an entity with the simplified temporal representation	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity with the simplified temporal representation	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: options set to '\$ {options}' and Query Parameter: context set to 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/'	SUT <- Client

	<pre> jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-10-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_12	
Test objective	Check that one can retrieve the temporal evolution of an entity with a LanguageProperty property	
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.5.18, 4.5.7, 4.5.9, 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_LANG and PICS_P_TEMPORAL_REP and PICS_P_SIMPTEMPORAL_REP_AND_R_SIMPTEMPORAL_REP and PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_12_01	
Name	020_12_01 Retrieve the normalized temporal representation of an entity with a LanguageProperty property	
Parent Release	v1.4.1	
Description	Check that one can retrieve the temporal evolution of an entity with a LanguageProperty property	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- language-property-normalized-temporal-representation-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_12_02	

Name	020_12_02 Retrieve the simplified temporal representation of an entity with a LanguageProperty property	
Parent Release	v1.4.1	
Description	Check that one can retrieve the temporal evolution of an entity with a LanguageProperty property	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: options set to 'temporalValues' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- language-property-simplified-temporal-representation-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_07	
Test objective	Check that one cannot retrieve the temporal evolution of a non-existing entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_07_01	
Name	020_07_01 Retrieve the temporal evolution of a non-existing entity	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve the temporal evolution of a non-existing entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to 'urn:ngsi- ld:Vehicle:unknowEntity' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 404 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and </pre>	SUT -> Client

	Retrieve Temporal Representation Of Entity with Response body containing 'title' element }	
--	---	--

TP Id	TP/NGSI-LD/CI/Cons/TE/020_05	
Test objective	Check that one can retrieve the temporal evolution of the last N instances of entity attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${attrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_05_01	
Name	020_05_01 Retrieve Some Instances	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of the last N instances of entity attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: lastN set to '10' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-05-01-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_05_02	
Name	020_05_02 Retrieve All Instances	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of the last N instances of entity attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: lastN set to '20' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-05-02-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/TE/020_01	
Test objective	Check that one can retrieve the temporal evolution of an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_01_01	
Name	020_01_01 Retrieve the temporal evolution of an entity	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-id/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-01-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_03	
Test objective	Check that one can retrieve the temporal evolution of certain attributes of an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${atrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_03_01	

Name	020_03_01 Retrieve the temporal evolution of certain attributes of an entity	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of certain attributes of an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attrs set to '\$ {temporal_attributes_to_be_retrieved}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-03-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_04	
Test objective	Check that one can retrieve the temporal evolution of an entity matching the given NGSI-LD temporal query	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${attrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_04_01	
Name	020_04_01 After	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity matching the given NGSI-LD temporal query	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-08-01T13:03:00Z' and Query Parameter: endTimeAt set to '\$ {EMPTY}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-04-01-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_04_02	
Name	020_04_02 Before	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity matching the given NGSI-LD temporal query	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2020-08-01T12:05:00Z' and Query Parameter: endTimeAt set to '\$ {EMPTY}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-04-02-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_04_03	
Name	020_04_03 Between	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity matching the given NGSI-LD temporal query	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and </pre>	SUT <- Client

	<pre> Query Parameter: timerel set to 'between' and Query Parameter: timeAt set to '2020-08-01T12:00:00Z' and Query Parameter: endTimeAt set to '2020-08-01T13:00:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-04-03-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_06	
Test objective	Check that one cannot retrieve the temporal evolution of an entity with an invalid id (invalid URI)	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_06_01	
Name	020_06_01 Retrieve the temporal evolution of an entity with an invalid id	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve the temporal evolution of an entity with an invalid id (invalid URI)	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to 'invalidUri' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 400 and Retrieve Temporal Representation Of Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Retrieve Temporal Representation Of Entity with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/020_02	
Test objective	Check that one can retrieve the temporal evolution of an entity using a context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.3	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntRet	
Initial conditions	with { the SUT containing an initial state }	

Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/020_02_01	
Name	020_02_01 Retrieve the temporal evolution of an entity using a context	
Parent Release	v1.3.1	
Description	Check that one can retrieve the temporal evolution of an entity using a context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'GET' Retrieve Temporal Representation of Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Temporal Representation Of Entity with Response Status Code set to 200 and Retrieve Temporal Representation Of Entity with Response Body containing EntityTemporal element containing attribute instances in the time range specified by the NGSI- LD temporal query: * the payload is defined in the file set to 'vehicle- temporal-representation-020-02-expectation.jsonld' * the id was changed to '\$ {temporal_entity_representation_id}' * response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client

4.4.1.1.2 QueryTemporalEvolutionOfEntities

TP Id	TP/NGSI-LD/CI/Cons/TE/021_13	
Test objective	Check that one can query the temporal evolution of entities using the entityOperations method	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_ QueTemEvoEnt_AND_ QueTemEvoEntPos	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an a list of Temporal Entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_13_01	
Name	021_13_01 After	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities using the entityOperations method	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entityOperations/query' method set to 'POST' Query Temporal Representation of Entities Via POST Request: and Query Parameter: query_file_name set to 'entity- operations-after-query.jsonld' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: content_type set to 'application/ </pre>	SUT <- Client

	<pre> json' } then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities Via Post with Response Status Code set to 200 and Query Temporal Representation Of Entities Via Post with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-13-01-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_13_02	
Name	021_13_02 Before	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities using the entityOperations method	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entityOperations/query' method set to 'POST' Query Temporal Representation of Entities Via POST Request: and Query Parameter: query_file_name set to 'entity- operations-before-query.jsonld' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: content_type set to 'application/ json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities Via Post with Response Status Code set to 200 and Query Temporal Representation Of Entities Via Post with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-13-02-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_08
Test objective	Check that one can query the temporal evolution of entities matching the given NGSI-LD query
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4
Config Id	CF_01
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_08_01
Name	021_08_01 Query the temporal evolution of entities matching the given NGSI-LD query
Parent Release	v1.3.1
Description	Check that one can query the temporal evolution of entities matching the given NGSI-LD query

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: ngsild_query set to 'speed> 90' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-08-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_11	
Test objective	Check that one can query the temporal evolution of entities with a limit to the number of entities to be retrieved	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an a list of Temporal Entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_11_01	
Name	021_11_01 Query Some entities	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities with a limit to the number of entities to be retrieved	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: limit set to '\${2}' ' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Response Body </pre>	SUT -> Client

	containing a list of entities equal to '{\$2}' of type 'Vehicle' with response set to '\$ {response.json()}'	
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_11_02	
Name	021_11_02 Query All entities	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities with a limit to the number of entities to be retrieved	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: limit set to '{\$3}' ' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Response Body containing a list of entities equal to '{\$3}' ' of type 'Vehicle' with response set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_14	
Test objective	Check that one can query the temporal evolution of entities with the simplified representation	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_14_01	
Name	021_14_01 After	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities with the simplified representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-08-01T12:04:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: options set to 'temporalValues' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-14-01-expectation.json' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_14_02	
Name	021_14_02 Before	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities with the simplified representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2020-09-01T13:06:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: options set to 'temporalValues' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-14-02-expectation.json' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_09	
Test objective	Check that one can query the temporal evolution of entities matching the given NGSI-LD geo-query	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an a list of Temporal Entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_09_01	
Name	021_09_01 Near Point	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities matching the given NGSI-LD geo-query	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and </pre>	SUT <- Client

	<p>Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and</p> <p>Query Parameter: georel set to 'near;maxDistance==2000' and</p> <p>Query Parameter: geometry set to 'Point' and</p> <p>Query Parameter: coordinates set to '[-8.503,41.202]' and</p> <p>Query Parameter: geoproperty set to '\$ {EMPTY}' and</p> <p>Query Parameter: timerel set to 'after' and</p> <p>Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and</p> <p>Query Parameter: context set to 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/jsonld-contexts/ngsi-ld-test-suite-compound.jsonld'</p>	
	<p>then {</p> <p>the SUT sends a valid Response for the operations:</p> <p>Query Temporal Representation Of Entities with Response Status Code set to 200 and</p> <p>Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles-temporal-representation-021-09-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}'</p>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_09_02	
Name	021_09_02 Within Polygon	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities matching the given NGSI-LD geo-query	
Expected behaviour	Test events	Direction
	<p>when {</p> <p>the SUT receives a Request from the client containing:</p> <p>URL set to '/ngsi-ld/v1/temporal/entities'</p> <p>method set to 'GET'</p> <p>Query Temporal Representation of Entities and</p> <p>Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and</p> <p>Query Parameter: georel set to 'contains' and</p> <p>Query Parameter: geometry set to 'Polygon' and</p> <p>Query Parameter: coordinates set to '[[[-13.503,47.202],[6.541, 52.961],[20.37,44.653],[9.46,32.57],[-13.503,32.57],[-13.503,47.202]]]' and</p> <p>Query Parameter: geoproperty set to 'location' and</p> <p>Query Parameter: timerel set to 'after' and</p> <p>Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and</p> <p>Query Parameter: context set to 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/jsonld-contexts/ngsi-ld-test-suite-compound.jsonld'</p>	SUT <- Client
	<p>then {</p> <p>the SUT sends a valid Response for the operations:</p> <p>Query Temporal Representation Of Entities with Response Status Code set to 200 and</p> <p>Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles-temporal-representation-021-09-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}'</p>	SUT -> Client

j	TP/NGSI-LD/CI/Cons/TE/021_10	
Test objective	Check that one can query the temporal evolution of entities matching the given NGSI-LD Context Source filter	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT containing a Context Source Registration of a context source (CS1) providing temporal information of two entities of type Building between 2020-08-01T22:07:00Z and 2021-08-01T21:07:00Z and CS1 containing two temporal entities of type Building and temporal evolution of those entities in the mentioned interval. }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_10_01	
Name	021_10_01 Query the temporal evolution of entities matching the given NGSI-LD context source filter	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities matching the given NGSI-LD Context Source filter	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: csf set to 'endpoint' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Query Temporal Representation Of Entities with Response Status Code set to 200 } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_12	
Test objective	Check that one cannot query the temporal evolution of entities with an invalid request or invalid request content	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_12_01	
Name	021_12_01 Query the temporal evolution of entities with an invalid request	
Parent Release	v1.3.1	
Description	Check that one cannot query the temporal evolution of entities with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 400 and Query Temporal Representation Of Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Temporal Representation Of Entities with Response body containing 'title' element } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/Cons/TE/021_07	
Test objective	Check that one can query the temporal evolution of entities matching the given id pattern	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_07_01	
Name	021_07_01 Query the temporal evolution of entities matching the given id pattern	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities matching the given id pattern	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: entity_id_pattern set to 'urn:ngsi- ld:Vehicle:.*' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-07-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_05	
Test objective	Check that one can query the temporal evolution of entities matching the given type(s)	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_05_01	
Name	021_05_01 Query the temporal evolution of entities matching the given type(s)	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities matching the given type(s)	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-05-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_01	
Test objective	Check that one can query the temporal evolution of entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_01_01	
Name	021_01_01 After	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-08-01T12:04:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-01-01-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_01_02	

Name	021_01_02 Before	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'before' and Query Parameter: timeAt set to '2020-09-01T13:06:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-01-02-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_03	
Test objective	Check that one can query the temporal evolution of the last N instances of entities attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_QueTemEvoEnt_AND_QueTemEvoEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_03_01	
Name	021_03_01 Query the temporal evolution of the last N instances of entities attributes	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of the last N instances of entities attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: lastN set to '14' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- </pre>	SUT -> Client

	<pre>{entity_ids_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</pre>	
	<pre>then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-06-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' }</pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/TE/021_02	
Test objective	Check that one can query the temporal evolution of certain attributes of entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.4	
Config Id	CF_01	
PICS Selection	PICS_ QueTemEvoEnt_AND_ QueTemEvoEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/TE/021_02_01	
Name	021_02_01 Query the temporal evolution of certain attributes of entities	
Parent Release	v1.3.1	
Description	Check that one can query the temporal evolution of certain attributes of entities	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'GET' Query Temporal Representation of Entities and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: timerel set to 'after' and Query Parameter: timeAt set to '2020-07-01T12:05:00Z' and Query Parameter: attrs set to '\$ {temporal_attributes_to_be_retrieved}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: Query Temporal Representation Of Entities with Response Status Code set to 200 and Query Temporal Representation Of Entities with Request response body containing a list that contains Entity Temporal Elements compared with file 'vehicles- temporal-representation-021-02-expectation.jsonld' and using the list of entity ids define in '\$ {temporal_entities_representation_ids}' }</pre>	SUT -> Client

4.4.1.2 Discovery

4.4.1.2.1 RetrieveAvailableEntityTypes

TP Id	TP/NGSI-LD/CI/Cons/DISC/022_01	
Test objective	Check that one can retrieve a list of NGSI-LD entity types	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.5	
Config Id	CF_01	
PICS Selection	PICS_RetAvaEntTyp	
Initial conditions	with { the SUT being in the "initial state" and containing a list of entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/022_01_01	
Name	022_01_01 WithoutJsonLdContext	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list of NGSI-LD entity types	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/types' method set to 'GET' Retrieve entity types: and Query Parameter: context set to '{\$EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Entity Types with Response Status Code set to 200 and Retrieve Entity Types with Response Body containing an Entity Type List with expectation body equal to file: 'types/expectations/entity-type-list-022-01-01-expectation.json' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/022_01_02	
Name	022_01_02 WithJsonLdContext	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list of NGSI-LD entity types	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/types' method set to 'GET' Retrieve entity types: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Entity Types with Response Status Code set to 200 and Retrieve Entity Types with Response Body containing an Entity Type List with expectation body equal to file: 'types/expectations/entity-type-list-022-01-02-expectation.json' }	SUT -> Client

4.4.1.2.2 RetrieveAvailableAttributes

TP Id	TP/NGSI-LD/CI/Cons/DISC/025_01
Test objective	Check that one can retrieve a list of NGSI-LD attributes
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.8
Config Id	CF_01
PICS Selection	PICS_AvaAtt
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }

Permutations							
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/025_01_01						
Name	025_01_01 WithoutJsonLdContext						
Parent Release	v1.3.1						
Description	Check that one can retrieve a list of NGSI-LD attributes						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to '{\$EMPTY}' } </td> <td>SUT <- Client</td> </tr> <tr> <td> then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-01-expectation.json' * response set to '\$ {response.json()}' } </td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to '{\$EMPTY}' }	SUT <- Client	then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-01-expectation.json' * response set to '\$ {response.json()}' }	SUT -> Client
Test events	Direction						
when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to '{\$EMPTY}' }	SUT <- Client						
then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-01-expectation.json' * response set to '\$ {response.json()}' }	SUT -> Client						
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/025_01_02						
Name	025_01_02 WithJsonLdContext						
Parent Release	v1.3.1						
Description	Check that one can retrieve a list of NGSI-LD attributes						
Expected behaviour	<table border="1"> <thead> <tr> <th>Test events</th> <th>Direction</th> </tr> </thead> <tbody> <tr> <td> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </td> <td>SUT <- Client</td> </tr> <tr> <td> then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-02-expectation.json' * response set to '\$ {response.json()}' } </td> <td>SUT -> Client</td> </tr> </tbody> </table>	Test events	Direction	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client	then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-02-expectation.json' * response set to '\$ {response.json()}' }	SUT -> Client
Test events	Direction						
when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	SUT <- Client						
then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response Body containing an Attribute List element * with filename set to 'types/expectations/attribute-list-025-01-02-expectation.json' * response set to '\$ {response.json()}' }	SUT -> Client						

4.4.1.2.3 RetrieveDetailsOfAvailableEntityTypes

TP Id	TP/NGSI-LD/CI/Cons/DISC/023_01
Test objective	Check that one can retrieve a list with a detailed representation of NGSI-LD entity types
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.6
Config Id	CF_01
PICS Selection	PICS_RetAvaEntTyp
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/023_01_01
Name	023_01_01 WithoutJsonLdContext
Parent Release	v1.3.1
Description	Check that one can retrieve a list with a detailed representation of NGSI-LD entity types

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/types' method set to 'GET' Retrieve entity types: and Query Parameter: context set to '{\$EMPTY}' and Query Parameter: details set to 'true' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Entity Types with Response Status Code set to 200 and Retrieve Entity Types with Response Body containing an Entity Type Element with expectation body equal to file: 'types/expectations/entity-type-023-01-01-expectation.json' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/023_01_02	
Name	023_01_02 WithJsonLdContext	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list with a detailed representation of NGSI-LD entity types	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/types' method set to 'GET' Retrieve entity types: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: details set to 'true' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Entity Types with Response Status Code set to 200 and Retrieve Entity Types with Response Body containing an Entity Type Element with expectation body equal to file: 'types/expectations/entity-type-023-01-02-expectation.json' }	SUT -> Client

4.4.1.2.4 RetrieveAvailableAttributeInformation

TP Id	TP/NGSI-LD/CI/Cons/DISC/027_02	
Test objective	Check that one can retrieve a list with a detailed representation of NGSI-LD attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.10	
Config Id	CF_01	
PICS Selection	PICS_DetAvaAtt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/027_02_01	
Name	027_02_01 Retrieve Detailed Representation Of Available Attribute	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list with a detailed representation of NGSI-LD attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/attributes/{attribute_name}' method set to 'GET' Retrieve Attribute with attributeName set to 'airQualityLevel' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Attribute with Response Status Code set to 200 and Retrieve Attribute with Response body containing an array of attributes * with the expected payload defined in the file 'types/ expectations/attribute-027-01-expectation.json' * and response body set to '\$ {response.json()}' } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/DISC/027_01	
Test objective	Check that one cannot retrieve a detailed representation of an unknown NGSI-LD attribute	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.10	
Config Id	CF_01	
PICS Selection	PICS_DetAvaAtt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/027_01_01	
Name	027_01_01 Retrieve Detailed Representation Of Available Attribute Without Context	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a detailed representation of an unknown NGSI-LD attribute	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/attributes/{attribute_name}' method set to 'GET' Retrieve Attribute with attributeName set to 'airQualityLevel' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Attribute with Response Status Code set to 404 and Retrieve Attribute with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Attribute with Response body containing 'title' element } </pre>	SUT -> Client

4.4.1.2.5 RetrieveDetailsOfAvailableAttributes

TP Id	TP/NGSI-LD/CI/Cons/DISC/026_01	
Test objective	Check that one can retrieve a list with a detailed representation of NGSI-LD attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.9	
Config Id	CF_01	
PICS Selection	PICS_AvaAtt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/026_01_01	
Name	026_01_01 WithoutJsonLdContext	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list with a detailed representation of NGSI-LD attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to '\${EMPTY}' and Query Parameter: details set to 'true' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response body containing an array of attributes * with the expected payload defined in the file 'types/ expectations/attribute-026-01-01-expectation.json' * and response body set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/026_01_02	
Name	026_01_02 WithJsonLdContext	
Parent Release	v1.3.1	
Description	Check that one can retrieve a list with a detailed representation of NGSI-LD attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/attributes' method set to 'GET' Retrieve attributes: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: details set to 'true' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Attributes with Response Status Code set to 200 and Retrieve Attributes with Response body containing an array of attributes * with the expected payload defined in the file 'types/ expectations/attribute-026-01-02-expectation.json' * and response body set to '\$ {response.json()}' } </pre>	SUT -> Client

4.4.1.2.6 RetrieveAvailableEntityTypeInformation

TP Id	TP/NGSI-LD/CI/Cons/DISC/024_02	
Test objective	Check that one can retrieve a detailed representation of a specified NGSI-LD entity type	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.7	
Config Id	CF_01	
PICS Selection	PICS_DetAvaEntTyp	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/024_02_01	
Name	024_02_01 Retrieve Detailed Representation Of Available Entity Type	
Parent Release	v1.3.1	
Description	Check that one can retrieve a detailed representation of a specified NGSI-LD entity type	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/types/{type}' method set to 'GET' Retrieve Entity Type, with type set to 'Building', with Header['Link'] containing 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Retrieve Entity Type with Response Status Code set to 200 and Retrieve Entity Type with Response body containing an Entity Type Info with expectation body set to file 'types/ expectations/entity-type-info-024-01- expectation.json' and response body to be checked set to '\$ {response.json()}' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/Cons/DISC/024_01	
Test objective	Check that one cannot retrieve a detailed representation of an unknown NGSI-LD entity type	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.7	
Config Id	CF_01	
PICS Selection	PICS_DetAvaEntTyp	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/DISC/024_01_01	
Name	024_01_01 Retrieve Detailed Representation Of Available Entity Type Without Context	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a detailed representation of an unknown NGSI-LD entity type	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/types/{type}' method set to 'GET' Retrieve Entity Type, with type set to 'Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Entity Type with Response Status Code set to 404 and Retrieve Entity Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Retrieve Entity Type with Response body containing 'title' element } </pre>	SUT -> Client

4.4.1.3 Entity

4.4.1.3.1 QueryEntities

TP Id	TP/NGSI-LD/CI/Cons/E/019_03_04	
Test objective	Check that one cannot query entities if the requested attribute names are incorrect	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_03_04	
Name	019_03_04 Query several entities based on incorrect attribute names	
Parent Release	v1.3.1	
Description	Check that one cannot query entities if the requested attribute names are incorrect	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: attrs set to '\${attributes_to_be_retrieved}' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 400 and Query Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Entities with Response body containing 'title' element } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/E/019_02_04	
Test objective	Check that one can query several entities via POST Interaction based on attribute names	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_02_04	
Name	019_02_04 Query several entities via POST Interaction based on attribute names	
Parent Release	v1.3.1	
Description	Check that one can query several entities via POST Interaction based on attribute names	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/query' method set to 'POST' Get Entities Via POST Request: and Query Parameter: attrs set to '\${attributes_to_be_retrieved}' and Query Parameter: content_type set to 'application/json' and Query Parameter: accept set to 'application/json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities Via POST with Response Status Code set to 200 and Query Entities Via POST with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'vehicle- simple-attributes-core-context-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_retrieved' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_03_02	
Test objective	Check that one cannot query entities if the requested entity types are incorrect	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_03_02	
Name	019_03_02 Query entities based on incorrect entity types	
Parent Release	v1.3.1	
Description	Check that one cannot query entities if the requested entity types are incorrect	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 400 and Query Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Query Entities with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_02_02	
Test objective	Check that one can query several entities via POST Interaction based on the entity type	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_ QueEnt_AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_02_02	
Name	019_02_02 Query several entities via POST Interaction based on the entities types	
Parent Release	v1.3.1	
Description	Check that one can query several entities via POST Interaction based on the entity type	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-lid/v1/entityOperations/query' method set to 'POST' Get Entities Via POST Request: and Query Parameter: entity_types set to 'https:// ngsi-lid-test-suite/context#Vehicle' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: content_type set to 'application/ json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities Via POST with Response Status Code set to 200 and Query Entities Via POST with Response body containing a list containing entity elements with different types and Query Parameter: filename set to 'two- vehicles-expectation.jsonld' and Query Parameter: entities_representation_ids set to 'entities_ids_to_be_compared' and Query Parameter: response_body set to 'response.json()' and Query Parameter: ignore_core_context_version set to 'True' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_03_05	
Test objective	Check that one cannot query entities if the request has a wrong geometryProperty	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_ QueEnt_ AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_03_05	
Name	019_03_05 Query entities when the request has a wrong geometryProperty	
Parent Release	v1.3.1	
Description	Check that one cannot query entities if the request has a wrong geometryProperty	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: geoproperty set to 'location' and Query Parameter: georel set to 'invalid' and Query Parameter: geometry set to 'Point' and Query Parameter: coordinates set to '[-8.503,41.202] ' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 400 and Query Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Entities with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_02_05	
Test objective	Check that one can query entities via POST Interaction asking for a GeoJSON representation	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_ QueEnt_ AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_02_05	
Name	019_02_05 Query several entities via POST Interaction asking for a GeoJSON representation	
Parent Release	v1.3.1	
Description	Check that one can query entities via POST Interaction asking for a GeoJSON representation	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/query' method set to 'POST' Get Entities Via POST Request: and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: accept set to 'application/ geo+json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: content_type set to 'application/ json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities Via POST with Response Status Code set to 200 and Query Entities Via POST with Check Response Body Content and Query Parameter: expectation_filename set to 'vehicle- parking-simple-expectation.geojson' and Query Parameter: response_body set to 'response.json()' and Query Parameter: additional_ignored_path set to '\$ {EMPTY}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_08	
Test objective	Query entities with Entity Type Selection Language.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clauses 4.17 and 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_TYPESELECTLANG and PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_08_01	
Name	019_08_01 query with one type	
Parent Release	v1.5.1	
Description	Query entities with Entity Type Selection Language.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_selection}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body contains entities ids } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_08_02	
Name	019_08_02 query with the AND operator	
Parent Release	v1.5.1	
Description	Query entities with Entity Type Selection Language.	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_selection}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" }'	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body contains entities ids }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_08_03	
Name	019_08_03 query with the OR operator	
Parent Release	v1.5.1	
Description	Query entities with Entity Type Selection Language.	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_selection}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" }'	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body contains entities ids }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_08_04	
Name	019_08_04 different query with the OR operator	
Parent Release	v1.5.1	
Description	Query entities with Entity Type Selection Language.	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_selection}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" }'	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body contains entities ids } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_08_05	
Name	019_08_05 query with two operators	
Parent Release	v1.5.1	
Description	Query entities with Entity Type Selection Language.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\$ {entity_types_selection}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body contains entities ids } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_03_01	
Test objective	Check that one cannot query entities if the requested ids are incorrect	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QUEEnt_AND_QUEEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_03_01	
Name	019_03_01 Query entities based on incorrect ids	
Parent Release	v1.3.1	
Description	Check that one cannot query entities if the requested ids are incorrect	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_ids set to '\${entity_ids_to_be_retrieved}' and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 400 and Query Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and </pre>	SUT -> Client

	Query Entities with Response body containing 'title' element }	
--	---	--

TP Id	TP/NGSI-LD/CI/Cons/E/019_02_01	
Test objective	Check that one can query one entity via POST Interaction based on id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_02_01	
Name	019_02_01 Query one entity via POST Interaction based on id	
Parent Release	v1.3.1	
Description	Check that one can query one entity via POST Interaction based on id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/query' method set to 'POST' Get Entities Via POST Request: and Query Parameter: entity_ids set to '\$ {first_entity_id}' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' and Query Parameter: content_type set to 'application/ json' and Query Parameter: accept set to 'application/ json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entities Via POST with Response Status Code set to 200 and Query Entities Via POST with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- minimal-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: ignore_core_context_version set to 'False' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_03_03	
Test objective	Check that one cannot query entities if the requested id pattern is incorrect	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_03_03	
Name	019_03_03 Query several entities based on incorrect id pattern	
Parent Release	v1.3.1	
Description	Check that one cannot query entities if the requested id pattern is incorrect	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_id_pattern set to 'invalid_entity_id_pattern**' and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 400 and Query Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Entities with Response body containing 'title' element } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/E/019_02_03	
Test objective	Check that one can query several entities via POST Interaction based on the given id pattern	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_02_03	
Name	019_02_03 Query several entities via POST Interaction based on the given id pattern	
Parent Release	v1.3.1	
Description	Check that one can query several entities via POST Interaction based on the given id pattern	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/query' method set to 'POST' Get Entities Via POST Request: and Query Parameter: entity_id_pattern set to 'urn:ngsi- ld:Building:.*' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' and Query Parameter: content_type set to 'application/ json' and Query Parameter: accept set to 'application/ json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities Via POST with Response Status Code set to 200 and Query Entities Via POST with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- minimal-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_02	
Test objective	Check that one can query several entities based on the entities types	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	

Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_02	
Name	019_01_02 Query several entities based on the entities types	
Parent Release	v1.3.1	
Description	Check that one can query several entities based on the entities types	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: accept set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response body containing a list containing entity elements with different types and Query Parameter: filename set to 'two- types-vehicle-offstreetparking-expectation.jsonld' and Query Parameter: entities_representation_ids set to 'entities_ids_to_be_compared' and Query Parameter: response_body set to 'response.json()' and Query Parameter: ignore_core_context_version set to 'True' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_04	
Test objective	Check that the queried entities by Id can be returned in a simplified representation	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.7	
Config Id	CF_01	
PICS Selection	PICS_HTTP_SIMP_REPR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_04_01	
Name	019_04_01 Query entities in a simplified representation	
Parent Release	v1.3.1	
Description	Check that the queried entities by Id can be returned in a simplified representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_ids set to '\${entities_ids_to_be_retrieved}' and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: options set to 'keyValues' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set </pre>	SUT -> Client

	<pre> to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- simple-attributes-simplified-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	
--	--	--

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Test objective	Check that one can query several entities based on scopes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 4.19	
Config Id	CF_01	
PICS Selection	PICS_SCOPESELECTLANG	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryWithFullScope	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' Madrid/Gardens/ParqueNorte' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${2}' ' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryWithPlusMatching	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' Madrid/+/ParqueNorte' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${2}' ' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryWithHashMatching	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' CompanyA/#' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${1}' ' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryNonEmptyScope	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' #' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${2}' ' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryWithAndScope	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' Madrid/Gardens/ParqueNorte;/CompanyA/OrganizationB/UnitC' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${1}' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_06	
Name	019_01_06 QueryWithOrScope	
Parent Release	v1.5.1	
Description	Check that one can query several entities based on scopes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: scopeq set to '/' Madrid/Gardens/ParqueNorte;/CompanyA/OrganizationB/UnitC' and Query Parameter: entity_types set to 'https:// ngsi-ld-test-suite/context#Building' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${2}' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_04	
Test objective	Check that one can query several entities based on attribute names	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_ QueEnt_AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_04	
Name	019_01_04 Query several entities based on attribute names	
Parent Release	v1.3.1	
Description	Check that one can query several entities based on attribute names	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to 'ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and	SUT <- Client

	<pre> Query Parameter: attrs set to '\${attributes_to_be_retrieved}' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- attributes-query-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_06	
Test objective	Check that one can query entities specifying a maximum number of results	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.10	
Config Id	CF_01	
PICS Selection	PICS_HTTP_PAGINATION_LIMIT	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_06_01	
Name	019_06_01 Query entities specifying a maximum number of results	
Parent Release	v1.3.1	
Description	Check that one can query entities specifying a maximum number of results	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_ids set to '\${entity_ids_to_be_retrieved}' and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: limit set to '2' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list of entities equal to '\${2}' ' of type 'https://ngsi-ld-test-suite/ context#Building' with response set to '\$ {response.json()}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_03	
Test objective	Check that one can query several entities based on the given id pattern	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	

Config Id	CF_01	
PICS Selection	PICS_ QueEnt_AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_03	
Name	019_01_03 Query several entities based on the given id pattern	
Parent Release	v1.3.1	
Description	Check that one can query several entities based on the given id pattern	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-Id/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_id_pattern set to 'urn:ngsi-Id:Building:.*' and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- minimal-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: response_body set to 'response.json()' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_01	
Test objective	Check that one can query several entities based on ids	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_ QueEnt_AND_ QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_01	
Name	019_01_01 Query several entities based on ids	
Parent Release	v1.3.1	
Description	Check that one can query several entities based on ids	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-Id/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_ids set to '\${entities_ids_to_be_retrieved}' and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set </pre>	SUT -> Client

	<pre> to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- minimal-expectation.json' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: response_body set to 'response.json()' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	
--	---	--

TP Id	TP/NGSI-LD/CI/Cons/E/019_01_05	
Test objective	Check that one can query entities based on a geoquery	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.2	
Config Id	CF_01	
PICS Selection	PICS_QueEnt_AND_QueEntPos	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_01_05	
Name	019_01_05 Query several entities based on a geoquery	
Parent Release	v1.3.1	
Description	Check that one can query entities based on a geoquery	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to '\${entity_types_to_be_retrieved}' and Query Parameter: accept set to 'application/ld+json' and Query Parameter: georel set to 'equals' and Query Parameter: geometry set to 'Point' and Query Parameter: coordinates set to '[13.3986, 52.5547]' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entities with Response Status Code set to 200 and Query Entities with Response Body containing a list containing Entity Elements and Query Parameter: expectation_filename set to 'building- geoproperty-query-expectation.jsonld' and Query Parameter: entities_ids set to 'entities_ids_to_be_compared' and Query Parameter: ignore_core_context_version set to 'False' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_07
Test objective	If the count parameter is set to true the special HTTP header NGSILD-Results-Count is set in the response and it has to contain the total number of matching results.
Reference	ETSI GS CIM 009 V1.5.1 [1], clauses 5.7.2, 6.3.13
Config Id	CF_01
PICS Selection	PICS_QueEnt_AND_QueEntPos and PICS_HTTP_COUNT_RESULTS
Initial conditions	with { the SUT containing an initial state }

Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_07_01	
Name	019_07_01 Check that the total number of matching results is returned if the count parameter is set to true and only the entity type is provided	
Parent Release	v1.3.1	
Description	If the count parameter is set to true the special HTTP header NGSILD-Results-Count is set in the response and it has to contain the total number of matching results.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to 'Building' and Query Parameter: q set to '\$ {q}' and Query Parameter: count set to '\$ {True}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to \${expected_status_code} and Query Entities with Response Header: NGSILD- Results-Count equals to 2 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_07_02	
Name	019_07_02 Check that the total number of matching results is returned if the count parameter is set to true and a q parameter is provided	
Parent Release	v1.3.1	
Description	If the count parameter is set to true the special HTTP header NGSILD-Results-Count is set in the response and it has to contain the total number of matching results.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_types set to 'Building' and Query Parameter: q set to '\$ {q}' and Query Parameter: count set to '\$ {True}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to \${expected_status_code} and Query Entities with Response Header: NGSILD- Results-Count equals to 1 } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/019_05
Test objective	Check that the queried entities by id can be returned in GeoJSON format
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.7

Config Id	CF_01	
PICS Selection	PICS_HTTP_SIMP_REPR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/019_05_01	
Name	019_05_01 Get an entity by id that can be returned in GeoJSON format	
Parent Release	v1.3.1	
Description	Check that the queried entities by id can be returned in GeoJSON format	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'GET' Get Entities Request: and Query Parameter: entity_ids set to '\$ {entities_ids_to_be_retrieved}' and Query Parameter: entity_types set to '\$ {entity_types_to_be_retrieved}' and Query Parameter: accept set to 'application/ geo+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entities with Response Status Code set to 200 and Query Entities with Check Response Body Content and Query Parameter: expectation_filename set to 'two- buildings-location-attribute-normalized.geojson' and Query Parameter: response_body set to 'response.json()' and Query Parameter: additional_ignored_path set to '\$ {EMPTY}' } </pre>	SUT -> Client

4.4.1.3.2 RetrieveEntity

TP Id	TP/NGSI-LD/CI/Cons/E/018_01_02	
Test objective	Check that one can query some attributes from an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_01_02	
Name	018_01_02 Query some attributes from an entity	
Parent Release	v1.3.1	
Description	Check that one can query some attributes from an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' and Query Parameter: attrs set to '\${attributes_to_be_retrieved}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to </pre>	SUT -> Client

	201 and Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- simple-attributes-query-expectation.jsonld' }	
--	--	--

TP Id	TP/NGSI-LD/CI/Cons/E/018_01_01	
Test objective	Check that one can get an entity by id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_01_01	
Name	018_01_01 Get an entity by id	
Parent Release	v1.3.1	
Description	Check that one can get an entity by id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- simple-attributes-expectation.jsonld' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_01_03	
Test objective	Check that one can query the geometry property from an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_01_03	
Name	018_01_03 Query the geometry property from an entity	
Parent Release	v1.3.1	
Description	Check that one can query the geometry property from an entity	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' and Query Parameter: geoproperty set to 'location' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- geoproperty-query-expectation.jsonld' } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/E/018_02	
Test objective	Check that one cannot get an entity with invalid id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_02_01	
Name	018_02_01 Get an entity if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot get an entity with invalid/missing id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to 'thisisaninvaliduri' and Query Parameter: accept set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 400 and Query Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Query Entity with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_03_02	
Test objective	Check that one cannot get an entity if an attribute is not known to the system	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_03_02	
Name	018_03_02 Get an entity if an attribute is not known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot get an entity if an attribute is not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' and Query Parameter: attrs set to '\${attributes_to_be_retrieved}' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entity with Response Status Code set to 404 and Query Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Query Entity with Response body containing 'title' element } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Cons/E/018_04	
Test objective	Check that the queried entity by Id can be returned in a simplified representation	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.7	
Config Id	CF_01	
PICS Selection	PICS_HTTP_SIMP_REPR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_04_01	
Name	018_04_01 Get an entity in a simplified representation	
Parent Release	v1.3.1	
Description	Check that the queried entity by Id can be returned in a simplified representation	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' and Query Parameter: options set to 'keyValues' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- simple-attributes-simplified-expectation.jsonld' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_06	
Test objective	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_06_01	
Name	018_06_01 EmptyJsonLdContext	
Parent Release	v1.3.1	
Description	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: Link set to '< \${EMPTY}>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" }'	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- simple-attributes-sample-expanded-expectation.json' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_06_02	
Name	018_06_02 CreationTimeJsonLdContext	
Parent Release	v1.3.1	
Description	Check that the JSON-LD @context is obtained from a Link header if present and that the default JSON-LD @context is used if not present	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" }'	SUT <- Client
	then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- simple-attributes-sample-compacted-expectation.json' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_03_01
Test objective	Check that one cannot get an entity if the entity id is not known to the system
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.7.1
Config Id	CF_01
PICS Selection	PICS_EntRet
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_03_01
Name	018_03_01 Get an entity if the Entity Id is not known to the system
Parent Release	v1.3.1

Description	Check that one cannot get an entity if the entity id or attributes are not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 404 and Query Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Query Entity with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_07	
Test objective	Check that one can retrieve an entity using Language Filter	
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.15 and 5.7.1	
Config Id	CF_01	
PICS Selection	PICS_LANGFILTER and PICS_EntRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_01	
Name	018_07_01 Retrieve an entity using a specific natural language	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: lang set to 'fr' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-fr-filter.jsonld' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_02	
Name	018_07_02 Retrieve an entity using multiple natural languages with no ranked preference	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: lang set to 'fr- CH,fr' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-fr-filter.jsonld' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_03	
Name	018_07_03 Retrieve an entity with any supported language	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: lang set to '*' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-any-language-filter.jsonld' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_04	
Name	018_07_04 Retrieve an entity using multiple natural languages with ranked preferences	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: lang set to 'fr- CH,fr;q=0.9,en;q=0.8,*;q=0.5' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-fr-filter.jsonld' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_05	
Name	018_07_05 Retrieve an entity using a specific natural language with simplified representation	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' and Query Parameter: options set to 'keyValues' and Query Parameter: lang set to 'fr' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-fr-filter-simplified.jsonld' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_07_06	
Name	018_07_06 Retrieve an entity with any supported language with simplified representation	
Parent Release	v1.4.1	
Description	Check that one can retrieve an entity using Language Filter	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" and Query Parameter: options set to 'keyValues' and Query Parameter: lang set to '*' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- language-property-any-language-filter-simplified.jsonld' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Cons/E/018_05	
Test objective	Check that the queried entity by id can be returned in a GeoJSON format	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.7	
Config Id	CF_01	
PICS Selection	PICS_HTTP_SIMP_REPR	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_05_01	
Name	018_05_01 Simplified	
Parent Release	v1.3.1	
Description	Check that the queried entity by id can be returned in a GeoJSON format	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ geo+json' and Query Parameter: options set to 'keyValues' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json" } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- location-attribute-simplified.geojson' } </pre>	SUT -> Client

TP Permutation id	TP/NGSI-LD/CI/Cons/E/018_05_02	
Name	018_05_02 Normalized	
Parent Release	v1.3.1	
Description	Check that the queried entity by id can be returned in a GeoJSON format	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'GET' Get Entity Request: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ geo+json' and Query Parameter: options set to '\$ {EMPTY}' and Query Parameter: Link set to '< \$https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/ resources/jsonld-contexts/ngsi-ld-test-suite- compound.jsonld>; rel="http://www.w3.org/ns/json- ld#context";type="application/ld+json"' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Entity with Response Status Code set to 200 and Query Entity with Response Body containing an entity element with id set to '\$ {entity_id}' and body content set to 'building- location-attribute-normalized.geojson' } </pre>	SUT -> Client

4.4.2 Subscription

4.4.2.1 UpdateSubscription

TP Id	TP/NGSI-LD/CI/SUB/029_01	
Test objective	Check that one cannot update a subscription: If the Subscription id is not present or it is not a valid URI, then an error of type BadRequestData shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_01_01	
Name	029_01_01 NotPresentId	
Parent Release	v1.3.1	
Description	Check that one cannot update a subscription: If the Subscription id is not present or it is not a valid URI, then an error of type BadRequestData shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-update- sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 405 and Update Subscription with Response Body containing the type '\$ </pre>	SUT -> Client

Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_07_01	
Name	029_07_01 ActiveTrue	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: If isActive is equal to true and expiresAt is not present, then status shall be updated to "active", if and only if, the previous value of status was different than "expired"	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-isActive-true-update-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'active' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_05	
Test objective	Check that one can update a subscription: Term to URI expansion of Attribute names shall be observed	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_05_01	
Name	029_05_01 Update Subscription With Term to Uri Expansion with Context	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: Term to URI expansion of Attribute names shall be observed	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Link'] contain the context 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/ subscription-vehicle-entities-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Response Body containing the same content defined in file 'subscriptions/ expectations/subscription-vehicle-	SUT -> Client

	sample.jsonld' with subscription id '\$ {subscription_id}' }	
TP Permutation id	TP/NGSI-LD/CI/SUB/029_05_02	
Name	029_05_02 Update Subscription With Term to Uri Expansion without Context	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: Term to URI expansion of Attribute names shall be observed	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Link'] contain the context 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/ subscription-vehicle-entities-sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Response Body containing the same content defined in file 'subscriptions/ expectations/subscription-vehicle-expanded-types- sample.jsonld' with subscription id '\$ {subscription_id}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_02	
Test objective	Check that one cannot update a subscription: If the NGSI-LD System does not know about the target Subscription, because there is no existing Subscription whose id (URI) is equivalent, an error of type ResourceNotFound shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_02_01	
Name	029_02_01 Update Unknown Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot update a subscription: If the NGSI-LD System does not know about the target Subscription, because there is no existing Subscription whose id (URI) is equivalent, an error of type ResourceNotFound shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-update- sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 404 and Update Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and </pre>	SUT -> Client

	Update Subscription with Response body containing 'title' element }	
--	--	--

TP Id	TP/NGSI-LD/CI/SUB/029_04	
Test objective	Check that one cannot update a subscription: Any attempt to remove (by setting them to null in the Fragment) mandatory properties of a Subscription (clause 5.2.12) shall result in an error of type BadRequestData	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_04_01	
Name	029_04_01 Update Subscription With Null Mandatory Properties	
Parent Release	v1.3.1	
Description	Check that one cannot update a subscription: Any attempt to remove (by setting them to null in the Fragment) mandatory properties of a Subscription (clause 5.2.12) shall result in an error of type BadRequestData	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-null-properties-update-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 400 and Update Subscription with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Update Subscription with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_06	
Test objective	Check that one can update a subscription: The implementation shall modify the target Subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_06_01	
Name	029_06_01 Update Subscription	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: The implementation shall modify the target Subscription	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Link'] contain the context 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-update-sample.json' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Check Updated Entity and Query Parameter: 'updated_resource' set to 'subscription' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/SUB/029_08	
Test objective	Check that one can update a subscription: If isActive is equal to true and expiresAt corresponds to a DateTime in the future, then status shall be updated to "active"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_08_01	
Name	029_08_01 ActiveTrueExpiresAt	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: If isActive is equal to true and expiresAt corresponds to a DateTime in the future, then status shall be updated to "active"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-isActive- expiresAt-update-sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'active' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_11	
Test objective	Check that one cannot update a subscription: If expiresAt is included but referring to a DateTime in the past, then a BadRequestData error shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_11_01	
Name	029_11_01 Update Subscription With ExpiresAt In The Past	
Parent Release	v1.3.1	
Description	Check that one cannot update a subscription: If expiresAt is included but referring to a DateTime in the past, then a BadRequestData error shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-expiresAt- </pre>	SUT <- Client

	update-sample.json' }	
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 400 and Update Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Update Subscription with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_09	
Test objective	Check that one can update a subscription: If isActive is equal to false and expiresAt is not present, then status shall be updated to "paused", if and only if, the previous value of status was different than "expired"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_09_01	
Name	029_09_01 Update Subscription Status To Paused	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: If isActive is equal to false and expiresAt is not present, then status shall be updated to "paused", if and only if, the previous value of status was different than "expired"	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file 'subscriptions/fragments/subscription-isActive-false- update-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'paused' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/029_10	
Test objective	Check that one can update a subscription: If only expiresAt is included and refers to a DateTime in the future, then status shall be updated to "active", if and only if the previous value of status was "expired"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.2	
Config Id	CF_01	
PICS Selection	PICS_SubUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/029_10_01	
Name	029_10_01 Activate Expired Subscription	
Parent Release	v1.3.1	
Description	Check that one can update a subscription: If only expiresAt is included and refers to a DateTime in the future, then status shall be updated to "active", if and only if the previous value of status was "expired"	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/subscriptions/{subscription_id}' method set to 'PATCH' Request Update Subscription and Header['Content-Type'] set to 'application/json' and Payload defined in file '\${subscription_update_fragment_file_path}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Subscription with Response Status Code set to 204 and Retrieve Subscription with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'status' and Query Parameter: response_body set to 'response1.json()' and Query Parameter: expected_attribute_value set to 'active' }	SUT -> Client

4.4.2.2 RetrieveSubscription

TP Id	TP/NGSI-LD/CI/SUB/030_02	
Test objective	Check that one cannot retrieve a subscription: If the identifier provided does not correspond to any existing subscription in the system then an error of type ResourceNotFound shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.3	
Config Id	CF_01	
PICS Selection	PICS_SubRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/030_02_01	
Name	030_02_01 Retrieve Unknown Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a subscription: If the identifier provided does not correspond to any existing subscription in the system then an error of type ResourceNotFound shall be raised	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to 'urn:ngsi-lid:Subscription:unknowSubscription' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Retrieve Subscription with Response Status Code set to 404 and Retrieve Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ ResourceNotFound' and Retrieve Subscription with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/030_01	
Test objective	Check that one cannot retrieve a subscription if the subscription Id is not a valid URI, then an error of type BadRequestData shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.3	
Config Id	CF_01	
PICS Selection	PICS_SubRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/030_01_01	

Name	030_01_01 InvalidId	
Parent Release	v1.3.1	
Description	Check that one cannot retrieve a subscription: If the subscription Id is not present or it is not a valid URI, then an error of type BadRequestData shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to 'InvalidUri' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Subscription with Response Status Code set to 400 and Retrieve Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Retrieve Subscription with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/030_03	
Test objective	Check that one can retrieve a subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.3	
Config Id	CF_01	
PICS Selection	PICS_SubRet	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/030_03_01	
Name	030_03_01 Retrieve Subscription	
Parent Release	v1.3.1	
Description	Check that one can retrieve a subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-lid/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '\$ {subscription_id}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-lid-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-lid-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Retrieve Subscription with Response Status Code set to 200 and Retrieve Subscription with Response Body containing the same content defined in file 'subscriptions/ expectations/subscriptions-030-03- expectation.json' with subscription id '\$ {subscription_id}' } </pre>	SUT -> Client

4.4.2.3 SubscriptionNotificationBehaviour

TP Id	TP/NGSI-LD/CI/SUB/046_16_02	
Test objective	If a subscription defines an entity type selection query, a notification shall be sent whenever an entity matches the query.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	

PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_16_02	
Name	046_16_02 Check that a notification is not sent if the entity type does not match the entity type selection	
Parent Release	v1.5.1	
Description	If a subscription defines an entity type selection query, a notification shall not be sent if the entity type does not match the query	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'vehicle-simple-attributes-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Waiting for no Notification data and Query Parameter: timeout set to '5' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_14	
Test objective	The Notification content shall be JSON-LD when endpoint.accept is set to JSON-LD	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_14_01	
Name	046_14_01 Check that a notification is sent as JSON-LD	
Parent Release	v1.3.1	
Description	The Notification content shall be JSON-LD when endpoint.accept is set to 'application/ld+json'	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Notification with The dictionary ` \$ {notification}[data][0] ' should contain the key '@context' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_10	
Test objective	The Notification content shall be JSON by default.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_10_01	
Name	046_10_01 Check that the notification is sent as JSON	

Parent Release	v1.3.1	
Description	The Notification shall be sent as JSON	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_09	
Test objective	A Notification shall be sent (as mandated by each concrete binding and including any optional endpoint.receiverInfo defined by clause 5.2.22) to the endpoint specified by the endpoint.uri member of the notification structure defined by clause 5.2.14	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_09_01	
Name	046_09_01 Check that a notification is sent to the endpoint	
Parent Release	v1.3.1	
Description	A Notification shall be sent (as mandated by each concrete binding and including any optional endpoint.receiverInfo defined by clause 5.2.22) to the endpoint specified by the endpoint.uri member of the notification structure defined by clause 5.2.1	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received The dictionary ` \$ {headers}' should contain the key 'X- Additional-Key' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_12	
Test objective	A Notification shall be sent (as mandated by each concrete binding and including any optional endpoint.info defined by clause 5.2.22) to the endpoint specified by the endpoint.uri member of the notification structure defined by clause 5.2.14	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_12_01	
Name	046_12_01 Check that lastNotification is updated	
Parent Release	v1.3.1	

Description	The status, lastNotification and lastSuccess members shall be updated with expected value and dates. This test will check these formats.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '\$ {subscription_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received Notification data: '\$ {notification_info}[0] [status]' equal to 'ok' and Notification received The dictionary '\$ {notification_info}[0] ' should contain the key 'status' and Notification received The dictionary '\$ {notification_info}[0] ' should contain the key 'lastNotification' and Notification received The dictionary '\$ {notification_info}[0] ' should contain the key 'lastSuccess' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_16_01	
Test objective	If a subscription defines an entity type selection query, a notification shall be sent whenever an entity matches the query.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_16_01	
Name	046_16_01 Check that a notification is sent with entity matching the entity type selection	
Parent Release	v1.5.1	
Description	If a subscription defines an entity type selection query, a notification shall be sent whenever an entity matches the query.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-simple- attributes-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ </pre>	SUT -> Client

	<pre>{notification}[data][0][id] equal to '\$ {entity_id}' and Notification received The dictionary '\$ {notification}' should contain the key 'data' and Notification received The variable '\$ {notification}[data]' should not be '\${EMPTY}' }</pre>	
--	--	--

TP Id	TP/NGSI-LD/CI/SUB/046_15	
Test objective	If a Subscription defines a throttling member, a Notification shall not be sent if the throttling specified (in seconds) has not elapsed yet.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_15_01	
Name	046_15_01 Check that a notification is not sent if the throttling has not elapsed yet	
Parent Release	v1.3.1	
Description	If a Subscription defines a throttling member, a Notification shall not be sent if the throttling specified (in seconds) has not elapsed yet.	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'airQualityLevel- fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }</pre>	SUT <- Client
	<pre>then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received Waiting for no Notification data and Query Parameter: timeout set to '5' }</pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_13	
Test objective	If the response to the notification request is different than 200 OK then implementations shall: Update notification.lastFailure with a timestamp representing the current date and time, update notification.status to "failed"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_13_01	
Name	046_13_01 Check that lastFailure and status are updated if a notification could not be sent	
Parent Release	v1.3.1	
Description	If the response to the notification request is different than 200 OK then implementations shall: Update notification.lastFailure with a timestamp representing the current date and time, update notification.status to "failed"	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and }</pre>	SUT <- Client

	<pre> Query Parameter: id set to '\$ {subscription_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received Notification data: '\$ {notification_info}[0] [status]' equal to 'failed' and Notification received The dictionary `'\$ {notification_info}[0] ' should contain the key 'status' and Notification received The dictionary `'\$ {notification_info}[0] ' should contain the key 'lastFailure' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_11	
Test objective	A Notification shall be sent (as mandated by each concrete binding and including any optional endpoint.info defined by clause 5.2.22) to the endpoint specified by the endpoint.uri member of the notification structure defined by clause 5.2.14	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_11_01	
Name	046_11_01 Check that timesSent is increased by one	
Parent Release	v1.3.1	
Description	The notification.timesSent member shall be incremented by one.	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{id}' method set to 'GET' Subscription Retrieve with the following data: and Query Parameter: id set to '\$ {subscription_id}' and Query Parameter: accept set to 'application/ ld+json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received Notification data: '\$ {response.json()[notification] [timesSent]' equal to '\${1} ' and Retrieve Subscription received The dictionary `'\$ {response.json()} [notification]' should contain the key 'timesSent' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_08	
Test objective	Check that a notification is only sent if and only if the status is active	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	

Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Subscription \$ <u>{subscription}</u> with an id set to \$ <u>{subscription_id}</u> }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_08_01	
Name	046_08_01 Check that a notification is sent with all attributes	
Parent Release	v1.3.1	
Description	The structure of the notification message shall be as mandated by clause 5.3.1. The absence of the notification.attributes member of a Subscription means that all Entity Attributes shall be included. All attributes are included	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'airQualityLevel- fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][name] [value]' equal to 'Eiffel Tower' and Notification received The dictionary ` \$ {notification}' should contain the key 'data' and Notification received The dictionary ` \$ {notification}[data][0] ' should contain the key 'almostFull' and Notification received The dictionary ` \$ {notification}[data][0] ' should contain the key 'airQualityLevel' and Notification received The dictionary ` \$ {notification}[data][0] ' should contain the key 'subCategory' and Notification received The variable ` \$ {notification}[data]' should not be `\${EMPTY}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/046_08_02	
Name	046_08_02 Check that a notification is sent with all attributes in simplified format	
Parent Release	v1.3.1	
Description	The structure of the notification message shall be as mandated by clause 5.3.1. The absence of the notification.attributes member of a Subscription means that all Entity Attributes shall be included If the notification.format member value is "keyValues" then a simplified representation of the entities (as mandated by clause 4.5.3) shall be provided	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and	SUT <- Client

	<pre> Query Parameter: fragment_filename set to 'airQualityLevel- fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0] [name]' equal to 'Eiffel Tower' and Notification received The dictionary '\$ {notification}' should contain the key 'data' and Notification received The variable '\$ {notification}[data]' should not be '{\$EMPTY}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_04	
Test objective	If a Subscription does not define a timeInterval member, the notification shall be sent whenever there is a change in the watched Attributes. The notification message shall include all the subscribed Entities that changed and that match (as mandated by clauses 4.9 and 4.10) the query and geoquery conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_04_01	
Name	046_04_01 Check that a notification is sent with all entities	
Parent Release	v1.3.1	
Description	The notification message shall include all the subscribed Entities that changed and that match (as mandated by clauses 4.9 and 4.10) the query and geoquery conditions	
Expected behaviour	Test events	Direction
	<p>The client at \${endpoint} receives a valid Notification containing a subscription with id set to \${subscriptionId} and status equals 'active' and timeInterval is set to \${timeInterval}' and watchedAttributes is 'Empty' and q is 'Empty' and geoQ is 'Empty' and with subscription.entity with type 'Building' and notification.endpoint.accept is 'application/ json' When the timeinterval is reached at \${timeInterval} seconds the SUT needs to send out a notification to the client sends a notification to the client every \${timeInterval} seconds</p>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][airQualityLevel] [value]' equal to '\${5}' and Notification received Notification data: '\$ {notification}[data][0][name] [value]' equal to 'Eiffel Tower' and Notification received The dictionary '\$ {notification}' should contain the key 'data' and </pre>	SUT -> Client

	Notification received The variable ` \$ {notification}[data]' should not be '\${EMPTY}' }	
--	---	--

TP Id	TP/NGSI-LD/CI/SUB/046_06	
Test objective	If a Subscription does not define a timeInterval member, the notification shall be sent whenever an entity matches the query defined in the subscription. The notification message shall include all the subscribed Entities that changed and that match (as mandated by clauses 4.9 and 4.10) the query and geoquery conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_06_01	
Name	046_06_01 Check that a notification is sent with all matching entities	
Parent Release	v1.3.1	
Description	only the subscribed Entities matching the query and watched attributes shall be included.	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'building-location-attribute.jsonld' }	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\${entity_id}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_02	
Test objective	If a Subscription defines a timeInterval member, a Notification shall be sent periodically, when the time interval (in seconds) specified in such value field is reached, regardless of Attribute changes.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_02_01	
Name	046_02_01 Check that a notification is sent on the timeInterval	
Parent Release	v1.3.1	
Description	If a Subscription defines a timeInterval member, a Notification shall be sent periodically, when the time interval (in seconds) specified in such value field is reached, regardless of Attribute changes.	
Expected behaviour	Test events	Direction
	The client at \$ {endpoint} receives a valid Notification containing a subscription with id set to \${subscriptionId} and status equals 'active' and timeInterval is set to '\$ {timeInterval}' and watchedAttributes is 'Empty' and q is 'Empty' and geoQ is 'Empty' and with subscription.entity with type 'Building' and notification.endpoint.accept is 'application/ json' When the timeinterval is reached at \$ {timeInterval} seconds the SUT needs to send out a notification to the client sends a notification to the client every \$ {timeInterval} seconds	SUT <- Client

	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '15' seconds and Notification received After waiting '15' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][name] [value]' equal to 'Eiffel Tower' and Notification received The dictionary '\$ {notification}' should contain the key 'data' and Notification received The variable '\$ {notification}[data]' should not be '\$ {EMPTY}' and Notification received The expression '\$ {notification}[data][0][airQualityLevel][value] == '4.0' or '\$ {notification}[data][0][airQualityLevel] [value] == '4' should be True } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/SUB/046_07	
Test objective	Check that a notification is only sent if and only if the status is active	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Subscription \${subscription} with an id set to \${subscription_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_07_01	
Name	046_07_01 Check notification structure	
Parent Release	v1.3.1	
Description	The structure of the notification message shall be as mandated by clause 5.3. Valid notification with attributes as stated above	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/046_07_02	
Name	046_07_02 Check correct attributes are included	
Parent Release	v1.3.1	
Description	The structure of the notification message shall be as mandated by clause 5.3. The Entity Attributes included (Properties or Relationships) shall be those specified by the notification.attributes member in the Subscription data type (clause 5.2.12).	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and </pre>	SUT <- Client

	<pre> Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received The dictionary '\$ {notification}[data][0] ' should contain the key 'airQualityLevel' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/046_07_03	
Name	046_07_03 Check URI expansion is observed	
Parent Release	v1.3.1	
Description	The structure of the notification message shall be as mandated by clause 5.3. URI expansion shall be observed (clause 5.5.7).	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received The dictionary '\$ {notification}[data][0] ' should contain the key 'https://ngsi-ld-test- suite/context#airQualityLevel' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_05	
Test objective	If a Subscription does not define a timeInterval member, the notification shall be sent whenever there is a change in the watched Attributes and the entity matches the q parameter. The notification message shall include all the subscribed Entities that changed and that match (as mandated by clauses 4.9 and 4.10) the query and geoquery conditions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_05_01	
Name	046_05_01 Check that a notification is sent with all entities	
Parent Release	v1.3.1	
Description	The notification message shall include all the subscribed Entities that changed and that match (as mandated by clauses 4.9 and 4.10) the query and geoquery conditions	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'airQualityLevel- fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client

	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][name] [value]' equal to 'Eiffel Tower' and Notification received The dictionary '\$ {notification}' should contain the key 'data' and Notification received The variable '\$ {notification}[data]' should not be '\$ {EMPTY}' and Notification received The expression '\$ {notification}[data][0][airQualityLevel][value]'=='5.0' or '\$ {notification}[data][0][airQualityLevel] [value]'=='5' should be True } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/SUB/046_01	
Test objective	Check that a notification is only sent if and only if the status is active	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_01_01	
Name	046_01_01 Check that a notification is only sent if status is active	
Parent Release	v1.3.1	
Description	Check that a notification is only sent if and only if the status is active	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'airQualityLevel- fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '10' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][airQualityLevel] [value]' equal to '\${5}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/046_03	
Test objective	A notification with all subscribed Entities will be included if query or geoquery are not defined.	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.6	
Config Id	CF_01	
PICS Selection	PICS_NOTIFICATIONS_SUB	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/046_03_01	
Name	046_03_01 Check that a notification is sent with all entities	
Parent Release	v1.3.1	
Description	A notification with all subscribed Entities will be included if query or geoquery are not defined.	
Expected behaviour	Test events	Direction
	The client at \$ {endpoint} receives a valid Notification containing a subscription with id set to \${subscriptionId} and status equals 'active' and timeInterval is set to '\$ {timeInterval}' and watchedAttributes is 'Empty' and q is 'Empty' and geoQ is 'Empty' and with subscription.entity with type 'Building' and notification.endpoint.accept is 'application/ json' When the timeinterval is reached at \${timeInterval} seconds the SUT needs to send out a notification to the client sends a notification to the client every \${timeInterval} seconds	SUT <- Client
	then { the client at '\$ {endpoint}' receives a valid Notification containing: Notification received After waiting '5' seconds and Notification received Notification data: '\$ {notification}[subscriptionId]' equal to '\$ {subscription_id}' and Notification received Notification data: '\$ {notification}[data][0][id]' equal to '\$ {entity_id}' and Notification received Notification data: '\$ {notification}[data][0][name]' [value] equal to 'Eiffel Tower' and Notification received The dictionary ` \$ {notification}' should contain the key 'data' and Notification received The variable ` \$ {notification}[data]' should not be '\$ {EMPTY}' and Notification received The expression ` \$ {notification}[data][0][airQualityLevel][value]== '5.0' or '\$ {notification}[data][0][airQualityLevel][value]== '5' should be True }	SUT -> Client

4.4.2.4 QuerySubscriptions

TP Id	TP/NGSI-LD/CI/SUB/031_02
Test objective	Check that one can query a list of subscriptions: Pagination logic shall be in place
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.4
Config Id	CF_01
PICS Selection	PICS_RetLisSub
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Subscription \${subscription} with an id set to \${subscription_id} }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/SUB/031_02_01
Name	031_02_01 Query Second Subscription
Parent Release	v1.3.1
Description	Check that one can query a list of subscriptions: Pagination logic shall be in place

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/ method set to 'GET' Query Subscription Request with data: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${1}' ' and Query Parameter: offset set to '\${1}' ' and Query Parameter: accept set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Subscriptions with Response Status Code set to 200 and Query Subscriptions with Response Body containing a list of entities equal to '\$ {expectation_subscription_number}' of type 'Subscription' with response set to '\$ {response.json()}' and Query Subscriptions with Response header with 'Prev' header equal to '</ ngsi-ld/v1/subscriptions? limit=1&offset=2>;rel="next";type="application/ ld+json"' and with 'Next' header equal to '\$ {response.headers}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/031_02_02	
Name	031_02_02 Query Last Subscription	
Parent Release	v1.3.1	
Description	Check that one can query a list of subscriptions: Pagination logic shall be in place	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/ method set to 'GET' Query Subscription Request with data: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${1}' ' and Query Parameter: offset set to '\${2}' ' and Query Parameter: accept set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Subscriptions with Response Status Code set to 200 and Query Subscriptions with Response Body containing a list of entities equal to '\$ {expectation_subscription_number}' of type 'Subscription' with response set to '\$ {response.json()}' and Query Subscriptions with Response header with 'Prev' header equal to '\$ {EMPTY}' and with 'Next' header equal to '\$ {response.headers}' } </pre>	SUT -> Client

TP Permutation id	TP/NGSI-LD/CI/SUB/031_02_03	
Name	031_02_03 Query All Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query a list of subscriptions: Pagination logic shall be in place	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'GET' Query Subscription Request with data: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' and Query Parameter: limit set to '\${15}' ' and Query Parameter: offset set to '\${0}' ' and Query Parameter: accept set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Subscriptions with Response Status Code set to 200 and Query Subscriptions with Response Body containing a list of entities equal to '\$ {expectation_subscription_number}' of type 'Subscription' with response set to '\$ {response.json()}' and Query Subscriptions with Response header with 'Prev' header equal to '\$ {EMPTY}' and with 'Next' header equal to '\$ {response.headers}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/031_01	
Test objective	Check that one can query a list of subscriptions	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.4	
Config Id	CF_01	
PICS Selection	PICS_RetLisSub	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/SUB/031_01_01	
Name	031_01_01 Query Subscriptions	
Parent Release	v1.3.1	
Description	Check that one can query a list of subscriptions	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'GET' Query Subscription Request with data: and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Query Subscriptions with Response Status Code set to 200 and Query Subscriptions with Response containing: * file set to 'subscriptions/ expectations/subscriptions-031-01-expectation.json' * id set to '\$ </pre>	SUT -> Client

	{subscription_ids}' * response set to '\$ {response.json()}' }	
--	---	--

4.4.2.5 DeleteSubscription

TP Id	TP/NGSI-LD/CI/SUB/032_02	
Test objective	Check that one cannot delete a subscription: If the subscription id provided does not correspond to any existing subscription in the system then an error of type ResourceNotFound shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.5	
Config Id	CF_01	
PICS Selection	PICS_SubDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/032_02_01	
Name	032_02_01 Delete Unknown Subscription	
Parent Release	v1.3.1	
Description	Check that one cannot delete a subscription: If the subscription id provided does not correspond to any existing subscription in the system then an error of type ResourceNotFound shall be raised	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'DELETE' Delete Subscription with id set to 'urn:ngsi-ld:Subscription:unknowSubscription' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Subscription with Response Status Code set to 404 and Delete Subscription with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/ResourceNotFound' and Delete Subscription with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/032_03	
Test objective	Check that one can delete a subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.5	
Config Id	CF_01	
PICS Selection	PICS_SubDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/032_03_01	
Name	032_03_01 Delete Subscription	
Parent Release	v1.3.1	
Description	Check that one can delete a subscription	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'DELETE' Delete Subscription with id set to '\$ {subscription_id}' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Delete Subscription with Response Status Code set to 204 and Retrieve Subscription with Response Status Code set to \${response1.status_code} } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/SUB/032_01	
Test objective	Check that one cannot delete a subscription: If the subscription Id is not present or it is not a valid URI, then an error shall be raised	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.5	
Config Id	CF_01	
PICS Selection	PICS_SubDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/032_01_01	
Name	032_01_01 NotPresentId	
Parent Release	v1.3.1	
Description	Check that one cannot delete a subscription: If the subscription Id is not present or it is not a valid URI, then an error shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'DELETE' Delete Subscription with id set to '\${EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Subscription with Response Status Code set to 405 and Delete Subscription with Response Body containing the type '\$ {EMPTY}' and Delete Subscription with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/032_01_02	
Name	032_01_02 InvalidId	
Parent Release	v1.3.1	
Description	Check that one cannot delete a subscription: If the subscription Id is not present or it is not a valid URI, then an error shall be raised	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/subscriptions/{subscription_id}' method set to 'DELETE' Delete Subscription with id set to 'InvalidUri' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Subscription with Response Status Code set to 400 and Delete Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Delete Subscription with Response body containing 'title' element } </pre>	SUT -> Client

4.4.2.6 CreateSubscription

TP Id	TP/NGSI-LD/CI/SUB/028_01	
Test objective	Check that one can create a subscription	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.1	
Config Id	CF_01	

PICS Selection	PICS_CreSub	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/028_01_01	
Name	028_01_01 Create Subscription	
Parent Release	v1.3.1	
Description	Check that one can create a subscription	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscriptions/ subscription-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Retrieve Subscription with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/028_03	
Test objective	Check that one cannot create a subscription with an invalid/empty id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.1	
Config Id	CF_01	
PICS Selection	PICS_CreSub	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/028_03_01	
Name	028_03_01 InvalidId	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with an invalid/empty id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscriptions/ subscription-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 400 and Create Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Subscription with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/028_03_02	
Name	028_03_02 EmptyId	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with an invalid/empty id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' </pre>	SUT <- Client

	Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'subscriptions/subscription-sample.jsonld' }	
	then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 400 and Create Subscription with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Create Subscription with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/SUB/028_05	
Test objective	Check that one cannot create a subscription with invalid throttling	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.1	
Config Id	CF_01	
PICS Selection	PICS_CreSub	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/028_05_01	
Name	028_05_01 ThrottlingAndTimeIntervalConjunction	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with invalid throttling	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/subscriptions/' method set to 'POST' Create Subscription Request with Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'subscription-invalid-throttling-timeInterval-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Subscription From File with Response Status Code set to 400 and Create Subscription From File with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Create Subscription From File with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/028_05_02	
Name	028_05_02 NegativeThrottling	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with invalid throttling	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/subscriptions/' method set to 'POST' Create Subscription Request with Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'subscription-invalid-negative-throttling-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Subscription From File with Response Status Code set to	SUT -> Client

	<pre> 400 and Create Subscription From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Subscription From File with Response body containing 'title' element } </pre>	
--	--	--

TP Id	TP/NGSI-LD/CI/SUB/028_02	
Test objective	Check that one cannot create a subscription with an invalid request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.1	
Config Id	CF_01	
PICS Selection	PICS_CreSub	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/028_02_01	
Name	028_02_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' Create Subscription Request with Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscription- invalid-json-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Subscription From File with Response Status Code set to 400 and Create Subscription From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Create Subscription From File with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/SUB/028_02_02	
Name	028_02_02 EmptyJson	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' Create Subscription Request with Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscription-empty- sample.jsonld' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Subscription From File with Response Status Code set to 400 and Create Subscription From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Subscription From File with Response body containing 'title' element } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/SUB/028_04	
Test objective	Check that one cannot create a subscription with an existing id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.8.1	
Config Id	CF_01	
PICS Selection	PICS_CreSub	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/SUB/028_04_01	
Name	028_04_01 Create a subscription with an id known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot create a subscription with an existing id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/subscriptions/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'subscriptions/ subscription-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Subscription with Response Status Code set to 409 and Create Subscription with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ AlreadyExists' and Create Subscription with Response body containing 'title' element } </pre>	SUT -> Client

4.4.3 Provision

4.4.3.1 TemporalEntity

4.4.3.1.1 UpdateTemporalRepresentationOfEntity

TP Id	TP/NGSI-LD/CI/Prov/TE/008_01	
Test objective	Check that one can update a temporal representation of an entity with simple temporal properties	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.11	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/008_01_01	
Name	008_01_01 Update a temporal representation of an entity with simple temporal properties	
Parent Release	v1.3.1	
Description	Check that one can update a temporal representation of an entity with simple temporal properties	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'vehicle- create-temporal-representation-sample.jsonld' and Query Parameter: content_type set to 'application/ ld+json' and Query Parameter: accept set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'updated_resource' set to 'temporal_entity_expectation_payload' and Query Parameter: 'response_body' set to 'response2.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

4.4.3.1.2 DeleteTemporalRepresentationOfEntity

TP Id	TP/NGSI-LD/CI/Prov/TE/009_01	
Test objective	Check that one can delete a temporal representation of an entity with simple temporal properties	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.16	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/009_01_01	
Name	009_01_01 Delete a temporal representation of an entity with simple temporal properties	
Parent Release	v1.3.1	
Description	Check that one can delete a temporal representation of an entity with simple temporal properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'DELETE' Delete Temporal Representation Of Entity With Returning Response with id set to '\$ {temporal_entity_representation_id}' } </pre>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Delete Temporal Representation Of Entity With Returning Response with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Response Status Code set to \${response1.status_code} } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/Prov/TE/009_03	
Test objective	Check that an error is raised if one deletes a temporal entity with a non-existing/invalid EntityId	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.16	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/009_03_01	
Name	009_03_01 Delete a temporal representation of an entity with an unknown entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes a temporal entity with a non-existing entity id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'DELETE' Delete Temporal Representation Of Entity With Returning Response with id set to '\$ {temporal_entity_id}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Temporal Representation Of Entity With Returning Response with Response Status Code set to 404 and Delete Temporal Representation Of Entity With Returning Response with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Delete Temporal Representation Of Entity With Returning Response with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TE/009_02	
Test objective	Check that an error is raised if one deletes a temporal entity with an empty/invalid EntityId	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.16	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/009_02_01	
Name	009_02_01 Delete a temporal representation of an entity with an empty entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes a temporal entity with an empty/invalid EntityId	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'DELETE' Delete Temporal Representation Of Entity With Returning Response with id set to '\$ </pre>	SUT <- Client

	{EMPTY}' }	
	then { the SUT sends a valid Response for the operations: Delete Temporal Representation Of Entity With Returning Response with Response Status Code set to 405 and Delete Temporal Representation Of Entity With Returning Response with Response Body containing the type '\$ {EMPTY}' and Delete Temporal Representation Of Entity With Returning Response with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/009_02_02	
Name	009_02_02 Delete a temporal representation of an entity with an invalid entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes a temporal entity with an empty/invalid EntityId	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to ' ngsi-ld/v1/temporal/entities/{temporal_entity_representation_id}' method set to 'DELETE' Delete Temporal Representation Of Entity With Returning Response with id set to 'invalidId' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Temporal Representation Of Entity With Returning Response with Response Status Code set to 400 and Delete Temporal Representation Of Entity With Returning Response with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Delete Temporal Representation Of Entity With Returning Response with Response body containing 'title' element }	SUT -> Client

4.4.3.1.3 CreateTemporalRepresentationOfEntity

TP Id	TP/NGSI-LD/CI/Prov/TE/007_02	
Test objective	Check that one cannot create a temporal entity with an empty/invalid json/id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.11	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntCre	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_02_01	
Name	007_02_01 Create a temporal entity with an invalid json	
Parent Release	v1.3.1	
Description	Check that one cannot create a temporal entity with an empty/invalid json/id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to ' ngsi-ld/v1/temporal/entities' method set to 'POST' Create Temporal Representation of Entity Selecting Content Type: and Query Parameter: filename set to 'vehicle- temporal-representation-invalid-json-sample.jsonId' and Query Parameter: content_type set to 'application/ Id+json' }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 400 and Create Temporal Representation Of Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Create Temporal Representation Of Entity Selecting Content Type with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_02_02	
Name	007_02_02 Create a temporal entity with an empty json	
Parent Release	v1.3.1	
Description	Check that one cannot create a temporal entity with an empty/invalid json/id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'POST' Create Temporal Representation of Entity Selecting Content Type: and Query Parameter: filename set to 'vehicle- temporal-representation-empty-json-sample.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 400 and Create Temporal Representation Of Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Create Temporal Representation Of Entity Selecting Content Type with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TE/007_01	
Test objective	Check that one can create a temporal representation of an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.11	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_01_01	
Name	007_01_01 Create a temporal representation of an entity	
Parent Release	v1.3.1	
Description	Check that one can create a temporal representation of an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'vehicle- create-temporal-representation-sample.jsonld' and Query Parameter: content_type set to 'application/ ld+json' and </pre>	SUT <- Client

	<pre> Query Parameter: accept set to '\$ {EMPTY}' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Retrieve Temporal Representation Of Entity with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_01_02	
Name	007_01_02 Create a temporal entity with no context	
Parent Release	v1.3.1	
Description	Check that one can create a temporal representation of an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'vehicle- create-temporal-representation-without-context-sample.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: accept set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Retrieve Temporal Representation Of Entity with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TE/007_03	
Test objective	Check that one cannot create a temporal entity with an empty/invalid json/id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.11	
Config Id	CF_01	
PICS Selection	PICS_TemEvoEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_03_01	
Name	007_03_01 Create a temporal entity with missing id	
Parent Release	v1.3.1	
Description	Check that one cannot create a temporal entity with an invalid @context	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {EMPTY}' and </pre>	SUT <- Client

	<p>Query Parameter: filename set to 'vehicle-temporal-representation-without-id-sample.jsonld' and</p> <p>Query Parameter: content_type set to 'application/ld+json' and</p> <p>Query Parameter: accept set to '\$ {EMPTY}'</p>	
	<p>then {</p> <p>the SUT sends a valid Response for the operation:</p> <p>Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 400</p>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TE/007_03_02	
Name	007_03_02 Create a temporal invalid URI	
Parent Release	v1.3.1	
Description	Check that one cannot create a temporal entity with an invalid @context	
Expected behaviour	Test events	Direction
	<p>when {</p> <p>the SUT receives a Request from the client containing:</p> <p>URL set to '/ngsi-ld/v1/temporal/entities'</p> <p>method set to 'POST'</p> <p>Create or Update Temporal Representation of Entity Selecting Content Type: and</p> <p>Query Parameter: temporal_entity_representation_id set to 'invalidld' and</p> <p>Query Parameter: filename set to 'vehicle-temporal-representation-sample.jsonld' and</p> <p>Query Parameter: content_type set to 'application/ld+json' and</p> <p>Query Parameter: accept set to '\$ {EMPTY}'</p>	SUT <- Client
	<p>then {</p> <p>the SUT sends a valid Response for the operation:</p> <p>Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 400</p>	SUT -> Client

4.4.3.2 BatchEntities

4.4.3.2.1 CreateBatchOfEntities

TP Id	TP/NGSI-LD/CI/Prov/BE/003_08
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5
Config Id	CF_01
PICS Selection	PICS_HTTP_CONTEXT_RESOLV
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_08_01
Name	003_08_01 Create a batch of one entity not containing a JSON-LD @context with a JSON-LD content type
Parent Release	v1.3.1
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 400 and Batch Create Entities with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Batch Create Entities with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_10	
Test objective	Check that one can create a batch of entities where two have the same id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.7	
Config Id	CF_01	
PICS Selection	PICS_BatEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_10_01	
Name	003_10_01 Create a batch of three valid entities where two have the same id	
Parent Release	v1.5.1	
Description	Check that one can create a batch of entities where two have the same id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to "" and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 207 and Batch Create Entities with Response body containing batch operation result set to '\$ {expected_batch_operation_result}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_09	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_09_01	
Name	003_09_01 Create a batch of one entity with a Link header and a JSON-LD content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 400 and Batch Create Entities with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Batch Create Entities with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_03	
Test objective	Check that one cannot create a batch of entities with an invalid request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.7	
Config Id	CF_01	
PICS Selection	PICS_BatEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_03_01	
Name	003_03_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot create a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Batch Entity Delete Request with operation set to 'create', Content-Type set to 'application/ld+json', and body set to 'batch/invalid-json-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/InvalidRequest' and Batch Request Entities From File with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_03_02	
Name	003_03_02 EmptyJson	
Parent Release	v1.3.1	
Description	Check that one cannot create a batch of entities with an invalid request	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Batch Entity Delete Request with operation set to 'create', Content- Type set to 'application/ ld+json', and body set to 'batch/empty- sample.jsonld } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Batch Request Entities From File with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_01	
Test objective	Check that one can create a batch of entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.7	
Config Id	CF_01	
PICS Selection	PICS_BatEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_01_01	
Name	003_01_01 MinimalEntity	
Parent Release	v1.3.1	
Description	Check that one can create a batch of entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to " and payload set to a list of entities to be created } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Batch Create Entities with Response Body set to an array of created entities ids and Query Entities with Check Created Resource Set To and Query Parameter: 'expected_resources' set to 'entities_to_be_created' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_01_02	
Name	003_01_02 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can create a batch of entities	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to '' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Batch Create Entities with Response Body set to an array of created entities ids and Query Entities with Check Created Resource Set To and Query Parameter: 'expected_resources' set to 'entities_to_be_created' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_01_03	
Name	003_01_03 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can create a batch of entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to '' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Batch Create Entities with Response Body set to an array of created entities ids and Query Entities with Check Created Resource Set To and Query Parameter: 'expected_resources' set to 'entities_to_be_created' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_01_04	
Name	003_01_04 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can create a batch of entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to '' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Batch Create Entities with Response Body set to an array of created entities ids and Query Entities with Check Created Resource Set To and Query Parameter: 'expected_resources' set to 'entities_to_be_created' and Query Parameter: 'response_body' set to 'response1.json()' and	SUT -> Client

	Query Parameter: 'ignored_keys' set to '\$ {None}' }	
--	--	--

TP Id	TP/NGSI-LD/CI/Prov/BE/003_05	
Test objective	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_05_01	
Name	003_05_01 Create a batch of one entity using the default context with JSON content type	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context and retrieve the entity without context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_05_02	
Name	003_05_02 Create a batch of one entity using the default context with JSON content type	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context and retrieve the entity with context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'ngsi- ld:default-context/almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_07	
Test objective	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_07_01	
Name	003_07_01 Create a batch of one entity using a JSON-LD @context obtained from the request payload	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and the response attribute should be compacted as one used the same context as provided when creating the entity	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_07_02	
Name	003_07_02 Create a batch of one entity using a JSON-LD @context obtained from the request payload	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and response attribute should not be compacted as one did not provide a context containing this term	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload set to a list of entities to be created }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'https:// ngsi-ld-test-suite/context#almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_02
Test objective	Check that one can create a batch of entities where some will succeed and others will fail
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.7
Config Id	CF_01

PICS Selection	PICS_BatEntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_02_01	
Name	003_02_01 Create a batch of two valid entities and one invalid entity	
Parent Release	v1.3.1	
Description	Check that one can create a batch of two valid entities and one invalid entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to '' and payload set to a list of entities to be created } then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 207 and Batch Create Entities with Response body containing batch operation result set to '\$ {expected_batch_operation_result}' and Query Entities with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_06	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_06_01	
Name	003_06_01 Create a batch of one entity containing a JSON-LD @context with a JSON content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-lid/v1/entityOperations/create' method set to 'POST' Request Header['Content- Type'] set to 'application/json' and payload set to a list of entities to be created } then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 400 and Batch Create Entities with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Batch Create Entities with Response body containing 'title' element } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/003_04	
Test objective	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_04_01	
Name	003_04_01 Create a batch of one entity using a provided Link header with JSON content type and retrieve the entity with context detail	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload set to a list of entities to be created } then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' } </pre>	SUT <- Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/003_04_02	
Name	003_04_02 Create a batch of one entity using a provided Link header with JSON content type and retrieve the entity without context detail	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/create' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload set to a list of entities to be created } then { the SUT sends a valid Response for the operations: Batch Create Entities with Response Status Code set to 201 and Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'https:// ngsi-ld-test-suite/context#almostFull' and Query Parameter: response_body set to 'response1.json()' } </pre>	SUT <- Client

4.4.3.2.2 DeleteBatchOfEntities

TP Id	TP/NGSI-LD/CI/Prov/BE/006_02
Test objective	Check that one can delete a batch of entities where some will succeed and others will fail
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.10
Config Id	CF_01

PICS Selection	PICS_BatEntDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/006_02_01	
Name	006_02_01 Delete a batch of non-existing and existing entities	
Parent Release	v1.3.1	
Description	Check that one can delete a batch of non-existing and existing entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/entityOperations/delete' method set to 'POST' Batch Delete Entities: and Query Parameter: entities_ids_to_be_deleted set to '@{entities_ids_to_be_deleted}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Delete Entities with Response Status Code set to 207 and \${response.json()} with Response body containing batch operation result set to '\$ {response1}' and Query Entities with Response body is empty } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/006_04	
Test objective	Check that one can delete a batch of entities with the same id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.10	
Config Id	CF_01	
PICS Selection	PICS_BatEntDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/006_04_01	
Name	006_04_01 Delete a batch of existing entities with the same id	
Parent Release	v1.5.1	
Description	Check that one can delete a batch entities with the same id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/entityOperations/delete' method set to 'POST' Batch Delete Entities: and Query Parameter: entities_ids_to_be_deleted set to '@{entities_ids_to_be_deleted}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Delete Entities with Response Status Code set to 207 and \${response.json()} with Response body containing batch operation result set to '\$ {response1}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/006_01	
Test objective	Check that one can delete a batch of entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.10	
Config Id	CF_01	
PICS Selection	PICS_BatEntDel	
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }	

Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/006_01_01	
Name	006_01_01 Delete a batch of entities	
Parent Release	v1.3.1	
Description	Check that one can delete a batch of entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/delete' method set to 'POST' Batch Delete Entities: and Query Parameter: entities_ids_to_be_deleted set to '@{entities_ids_to_be_deleted}' } then { the SUT sends a valid Response for the operations: Batch Delete Entities with Response Status Code set to 204 and Query Entities with Response body is empty } </pre>	SUT <- Client
		SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/006_03	
Test objective	Check that one cannot delete a batch of entities with an invalid request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.10	
Config Id	CF_01	
PICS Selection	PICS_BatEntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/006_03_01	
Name	006_03_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot delete a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/delete' method set to 'POST' Batch Entity Delete Request with operation set to 'delete', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- sample.jsonld } then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Batch Request Entities From File with Response body containing 'title' element } </pre>	SUT <- Client
		SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/006_03_02	
Name	006_03_02 EmptyJson	
Parent Release	v1.3.1	
Description	Check that one cannot delete a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/delete' method set to 'POST' Batch Entity Delete Request with operation set to 'delete', Content- </pre>	SUT <- Client

	Type set to 'application/ ld+json', and body set to 'batch/empty- sample.jsonld } then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Batch Request Entities From File with Response body containing 'title' element }	SUT -> Client
--	---	---------------

4.4.3.2.3 UpsertBatchOfEntities

TP Id	TP/NGSI-LD/CI/Prov/BE/004_06	
Test objective	Check that one cannot upsert a batch of entities with an invalid request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_06_01	
Name	004_06_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot upsert a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert' method set to 'POST' Batch Entity Delete Request with operation set to 'upsert', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- sample.jsonld } then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Batch Request Entities From File with Response body containing 'title' element }	SUT <- Client
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert' method set to 'POST' Batch Entity Delete Request with operation set to 'upsert', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- ld-sample.jsonld }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_06_02	
Name	004_06_02 InvalidJsonLd	
Parent Release	v1.3.1	
Description	Check that one cannot upsert a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert' method set to 'POST' Batch Entity Delete Request with operation set to 'upsert', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- ld-sample.jsonld }	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Batch Request Entities From File with Response body containing 'title' element } </pre>	SUT -> Client
--	--	---------------

TP Id	TP/NGSI-LD/CI/Prov/BE/004_04	
Test objective	Check that one can upsert a batch of entities with update option	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_04_01	
Name	004_04_01 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of entities with update option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@\${entities_to_be_upserted}' with update_option set to 'update' and Content- Type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_04_02	
Name	004_04_02 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of entities with update option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@\${entities_to_be_upserted}' with update_option set to 'update' and Content- Type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client

TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_04_03	
Name	004_04_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of entities with update option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'update' and Content- Type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/004_02	
Test objective	Check that one can upsert a batch of non-existing and existing entities where non-existing will be created and existing will be replaced	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_02_01	
Name	004_02_01 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing and existing entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content- Type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_02_02	
Name	004_02_02 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing and existing entities	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' </pre>	SUT <- Client

	Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json'	
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_02_03	
Name	004_02_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing and existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/004_05	
Test objective	Check that one can upsert a batch of entities where some will succeed and others will fail	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_05_01	
Name	004_05_01 Upsert a batch of two valid entities and one invalid entity	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of two valid entities and one invalid entity	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 207 and Batch Upsert Entities with Response body containing batch operation	SUT -> Client

	result set to '\$ {expected_batch_operation_result}' and Query Entities with Updated Entities set to '\$ {upserted_entities}' valid entities }	
--	--	--

TP Id	TP/NGSI-LD/CI/Prov/BE/004_07	
Test objective	Check that one can upsert a batch of entities where two have the same id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_07_01	
Name	004_07_01 Upsert a batch of three valid entities where two have the same id	
Parent Release	v1.5.1	
Description	Check that one can upsert a batch of where two have the same id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/004_03	
Test objective	Check that one can upsert a batch of existing entities and they will be replaced	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_03_01	
Name	004_03_01 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client

TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_03_02	
Name	004_03_02 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_03_03	
Name	004_03_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/004_01	
Test objective	Check that one can upsert a batch of non-existing entities and they will be created	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.8	
Config Id	CF_01	
PICS Selection	PICS_BatEntCreUpdUps	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_01_01	
Name	004_01_01 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-	SUT <- Client

	Type set to 'application/ld+json' }	
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_01_02	
Name	004_01_02 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/004_01_03	
Name	004_01_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can upsert a batch of non-existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/upsert?options=\${update_option}' method set to 'POST' Request batch upsert operation over entity from filename '@{entities_to_be_upserted}' with update_option set to 'replace' and Content-Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Upsert Entities with Response Status Code set to 201 and Batch Upsert Entities with Response Body set to an array of created entities ids and Query Entities with Updated Entities set to '\$ {entities_to_be_upserted}' valid entities }	SUT -> Client

4.4.3.2.4 UpdateBatchOfEntities

TP Id	TP/NGSI-LD/CI/Prov/BE/005_04
Test objective	Check that one cannot update a batch of entities with an invalid request
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.9
Config Id	CF_01

PICS Selection	PICS_BatEntUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_04_01	
Name	005_04_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot update a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update' method set to 'POST' Batch Entity Delete Request with operation set to 'update', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- sample.jsonld } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Batch Request Entities From File with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_04_02	
Name	005_04_02 InvalidJsonLd	
Parent Release	v1.3.1	
Description	Check that one cannot update a batch of entities with an invalid request	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update' method set to 'POST' Batch Entity Delete Request with operation set to 'update', Content- Type set to 'application/ ld+json', and body set to 'batch/invalid-json- ld-sample.jsonld } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Request Entities From File with Response Status Code set to 400 and Batch Request Entities From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Batch Request Entities From File with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/005_02
Test objective	Check that one can update a batch of entities with noOverwrite option
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.9
Config Id	CF_01
PICS Selection	PICS_BatEntUpd
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_02_01
Name	005_02_01 EntityWithExistingAttributes
Parent Release	v1.3.1

Description	Check that one can update a batch of entities with noOverwrite option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_02_02	
Name	005_02_02 EntityWithNewAttribute	
Parent Release	v1.3.1	
Description	Check that one can update a batch of entities with noOverwrite option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_02_03	
Name	005_02_03 EntityWithNewAndExistingAttribute	
Parent Release	v1.3.1	
Description	Check that one can update a batch of entities with noOverwrite option	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/005_03	
Test objective	Check that one can update a batch of entities where some will succeed and others will fail	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.9	
Config Id	CF_01	
PICS Selection	PICS_BatEntUpd	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_03_01	
Name	005_03_01 Update a batch of non-existing and existing entities	
Parent Release	v1.3.1	
Description	Check that one can update a batch of non-existing and existing entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 207 and Batch Update Entities with Response body containing batch operation result set to '\$ {expected_batch_operation_result}' and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/BE/005_01	
Test objective	Check that one can update a batch of entities	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.9	
Config Id	CF_01	
PICS Selection	PICS_BatEntUpd	
Initial conditions	with { the SUT being in the "initial state" and and containing a list of entities }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_01_01	
Name	005_01_01 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can update a batch of entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_01_02	
Name	005_01_02 EntityWithSimpleRelationships	
Parent Release	v1.3.1	
Description	Check that one can update a batch of entities	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/BE/005_01_03	
Name	005_01_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can update a batch of entities	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entityOperations/update?\${params_as_string}' method set to 'POST' Request batch update operation over entity from filename '@{entities_to_be_updated}' with overwrite_option set to '\$ {EMPTY}' and Content-Type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Batch Update Entities with Response Status Code set to 204 and Query Entities with Updated Entities set to '\$ {updated_entities}' valid entities }	SUT -> Client

4.4.3.3 EntityAttributes

4.4.3.3.1 DeleteEntityAttribute

TP Id	TP/NGSI-LD/CI/Prov/EA/013_02	
Test objective	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.5	
Config Id	CF_01	
PICS Selection	PICS_AttDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_02_01	
Name	013_02_01 Delete an attribute if the Entity Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$	SUT <- Client

	{EMPTY}' and Query Parameter: attributeld set to 'speed' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' }	
	then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 400 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_02_02	
Name	013_02_02 Delete an attribute if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to 'thisIsAnInvalidURI' and Query Parameter: attributeld set to 'speed' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 400 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_02_03	
Name	013_02_03 Delete an attribute if the Attribute Name is not present	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: attributeld set to '\$ {EMPTY}' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 405 }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/013_04
Test objective	Check that one can delete a scope from an entity
Reference	ETSI GS CIM 009 V1.5.1 [1], clauses 4.18 and 5.6.5
Config Id	CF_01
PICS Selection	PICS_SCOPES and PICS_AtDel
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }

Permutations

TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_04	
Name	013_04 Delete scope from an entity	
Parent Release	v1.5.1	
Description	Check that one can delete a scope from an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/' method set to 'GET' Request Retrieve Entity by Id and Query Parameter: id set to '\${entity_id}' and Query Parameter: accept set to 'application/json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Entity Attributes with Response Status Code set to 204 and Delete Entity Attributes with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/013_03	
Test objective	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.5	
Config Id	CF_01	
PICS Selection	PICS_AttDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_03_01	
Name	013_03_01 Delete an attribute when the Entity Id is not known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to / ngsi-lid/v1/entities/{entityId}/attrs/{attributeId}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {not_found_entity_id}' and Query Parameter: attributeId set to 'speed' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to 'false' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 404 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_03_02	
Name	013_03_02 Delete an attribute when the Entity does not contain the target attribute id	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to / ngsi-lid/v1/entities/{entityId}/attrs/{attributeId}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {valid_entity_id}' and </pre>	SUT <- Client

	<p>Query Parameter: attributeld set to 'notFound' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to 'false' }</p>	
	<p>then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 404 }</p>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_03_03	
Name	013_03_03 Delete an attribute when the Entity does not contain the target attribute with same datasetId	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute from an entity with invalid/missing ids	
Expected behaviour	Test events	Direction
	<p>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: attributeld set to 'speed' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to 'false' }</p>	SUT <- Client
	<p>then { the SUT sends a valid Response for the operation: Delete Entity Attributes with Response Status Code set to 404 }</p>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/013_01	
Test objective	Check that one can delete an attribute from an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.5	
Config Id	CF_01	
PICS Selection	PICS_AttDel	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_01_01	
Name	013_01_01 Delete an attribute with the default instance	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute from an entity	
Expected behaviour	Test events	Direction
	<p>when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: attributeld set to 'speed' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to '\$ {deleteAll}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</p>	SUT <- Client

	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Delete Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response2.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_01_02	
Name	013_01_02 Delete an attribute with the datasetId	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute from an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: attributeId set to 'speed' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to '\$ {deleteAll}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Delete Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response2.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/013_01_03	
Name	013_01_03 Delete all target attribute instances	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute from an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}?\${params_as_string}' method set to 'DELETE' Delete Entity Attributes: and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: attributeId set to 'speed' and Query Parameter: datasetId set to '\$ {datasetId}' and Query Parameter: deleteAll set to '\$ {deleteAll}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ </pre>	SUT <- Client

	<pre> jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Delete Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response2.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

4.4.3.3.2 PartialAttributeUpdate

TP Id	TP/NGSI-LD/CI/Prov/EA/012_02	
Test objective	Check that one cannot perform a partial update on an entity attribute with invalid/missing ids	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.4	
Config Id	CF_01	
PICS Selection	PICS_ParAttUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_02_01	
Name	012_02_01 Make a partial attribute update if the Entity Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {EMPTY}' and Query Parameter: Attributeld set to 'speed' and Query Parameter: fragment_filename set to 'vehicle- speed-equal-datasetid-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 400 and Partial Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Partial Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_02_02	
Name	012_02_02 Make a partial attribute update if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeld}' method set to 'PATCH' </pre>	SUT <- Client

	<p>Request Partial Update Entity Attributes and Query Parameter: entityId set to 'thisisaninvaliduri' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle-speed-equal-datasetid-fragment.jsonld' and Query Parameter: content_type set to 'application/ld+json'</p>	
	<pre>then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 400 and Partial Update Entity Attributes with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Partial Update Entity Attributes with Response body containing 'title' element }</pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_02_03	
Name	012_02_03 Make a partial attribute update if the Attribute type does not match	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle-speed-equal-datasetid-different-type-fragment.jsonld' and Query Parameter: content_type set to 'application/ld+json' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 400 and Partial Update Entity Attributes with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Partial Update Entity Attributes with Response body containing 'title' element }</pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_02_04	
Name	012_02_04 Make a partial attribute update if the entity fragment is empty	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute with invalid/missing ids	
Expected behaviour	Test events	Direction
	<pre>when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'empty-fragment.json' and Query Parameter: content_type set to 'application/ld+json' }</pre>	SUT <- Client
	<pre>then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 400 and</pre>	SUT -> Client

	Partial Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Partial Update Entity Attributes with Response body containing 'title' element }	
--	---	--

TP Id	TP/NGSI-LD/CI/Prov/EA/012_04	
Test objective	Check that one can perform a partial update on a LanguageProperty property	
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.5.18 and 5.6.4	
Config Id	CF_01	
PICS Selection	PICS_LANG and PICS_ParAttUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} and an attribute with an id set to \${attrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_04_01	
Name	012_04_01 Check that one can partially update a LanguageProperty property	
Parent Release	v1.4.1	
Description	Check that one can perform a partial update on a LanguageProperty property	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: AttributeId set to 'street' and Query Parameter: fragment_filename set to 'building- language-property-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/012_03	
Test objective	Check that one cannot perform a partial update on an entity attribute if the entity id or attribute is not known to the system	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.4	
Config Id	CF_01	
PICS Selection	PICS_ParAttUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_03_01	
Name	012_03_01 Partial update when the Entity Id is not known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute if the entity id or attribute is not known to the system	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and }	SUT <- Client

	<pre> Query Parameter: entityId set to '\$ {not_found_entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle- isparked-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	
	<pre> then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 404 and Partial Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Partial Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_03_02	
Name	012_03_02 Partial update when no default instance and no datasetId specified	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute if the entity id or attribute is not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle- speed-no-datasetid-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 404 and Partial Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Partial Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_03_03	
Name	012_03_03 Partial update when no instance with the datasetId specified	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute if the entity id or attribute is not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle- speed-unknown-datasetid-fragment.jsonld' and Query Parameter: content_type set to 'application/ </pre>	SUT <- Client

	Id+json' }	
	then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 404 and Partial Update Entity Attributes with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/ResourceNotFound' and Partial Update Entity Attributes with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_03_04	
Name	012_03_04 Partial update when the Attribute Name does not exist in the entity	
Parent Release	v1.3.1	
Description	Check that one cannot perform a partial update on an entity attribute if the entity id or attribute is not known to the system	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {valid_entity_id}' and Query Parameter: AttributeId set to 'isParked2' and Query Parameter: fragment_filename set to 'vehicle-isparked-fragment.jsonld' and Query Parameter: content_type set to 'application/Id+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 404 and Partial Update Entity Attributes with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/ResourceNotFound' and Partial Update Entity Attributes with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/012_01	
Test objective	Check that one can perform a partial update on an entity attribute	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.4	
Config Id	CF_01	
PICS Selection	PICS_ParAttUpd	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} and an attribute with an id set to \${atrId} }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_01_01	
Name	012_01_01 Check that one can partially update an attribute	
Parent Release	v1.3.1	
Description	Check that one can perform a partial update on an entity attribute	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {entity_id}' and	SUT <- Client

	<p>Query Parameter: AttributeId set to 'isParked' and Query Parameter: fragment_filename set to 'vehicle-isparked-fragment.jsonId' and Query Parameter: content_type set to 'application/Id+json'</p>	
	<pre> then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/012_01_02	
Name	012_01_02 Check that one can partially update an attribute by specifying the datasetId	
Parent Release	v1.3.1	
Description	Check that one can perform a partial update on an entity attribute	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Request Partial Update Entity Attributes and Query Parameter: entityId set to '\$ {entity_id}' and Query Parameter: AttributeId set to 'speed' and Query Parameter: fragment_filename set to 'vehicle- speed-equal-datasetid-fragment.jsonId' and Query Parameter: content_type set to 'application/ Id+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Partial Update Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

4.4.3.3 UpdateEntityAttributes

TP Id	TP/NGSI-LD/CI/Prov/EA/011_04	
Test objective	Check that one cannot update entity attributes with invalid request body	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.2	
Config Id	CF_01	
PICS Selection	PICS_UpdAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_04_01	
Name	011_04_01 Update entity attributes with invalid entity fragments	
Parent Release	v1.3.1	
Description	Check that one cannot update an attribute if the entity fragment is invalid	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and </pre>	SUT <- Client

	<p>Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to '\${fragment_filename}' and Query Parameter: content_type set to 'application/ld+json'</p>	
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 400 and Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/011_02	
Test objective	Check that one cannot update entity attributes with invalid/missing id or invalid request body	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.2	
Config Id	CF_01	
PICS Selection	PICS_UpdAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_02_01	
Name	011_02_01 Update an attribute if the Entity Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot update entity attributes with invalid/missing id or invalid request body	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {EMPTY}' and Query Parameter: fragment_filename set to 'vehicle- speed-two-datasetid-01-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 400 and Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_02_02	
Name	011_02_02 Update an attribute if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot update entity attributes with invalid/missing id or invalid request body	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to 'thisisaninvaliduri' and Query Parameter: fragment_filename set to 'vehicle- speed-two-datasetid-01-fragment.jsonld' and Query Parameter: content_type set to 'application/ </pre>	SUT <- Client

	<pre> Id+json' } then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 400 and Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
--	---	---------------

TP Id	TP/NGSI-LD/CI/Prov/EA/011_05	
Test objective	Check that one can update a scope in an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clauses 4.18 and 5.6.2	
Config Id	CF_01	
PICS Selection	PICS_SCOPES and PICS_UpdAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_05_01	
Name	011_05_01 Update scope to an entity already having a scope	
Parent Release	v1.5.1	
Description	Check that scope is replaced if entity already has a scope	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/' method set to 'GET' Request Retrieve Entity by Id and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 204 and Update Entity Attributes with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_05_02	
Name	011_05_02 Update scope to an entity not having a scope	
Parent Release	v1.5.1	
Description	Check that scope is not added if entity does not already have a scope	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/' method set to 'GET' Request Retrieve Entity by Id and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ </pre>	SUT <- Client

	jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }	
	then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 207 and Update Entity Attributes with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/011_01	
Test objective	Check that one can update entity attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.2	
Config Id	CF_01	
PICS Selection	PICS_UpdAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_01_01	
Name	011_01_01 Check that one can update existing attributes with no datasetId	
Parent Release	v1.3.1	
Description	Check that one can update entity attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- speed-two-datasetid-01-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 204 and Update Entity Attributes with Check Response Body Content and Query Parameter: expectation_filename set to 'expectation_resp_filename' and Query Parameter: response_body set to 'response.json()' and Query Parameter: additional_ignored_path set to '\$ {EMPTY}' and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_01_02	
Name	011_01_02 Check that one can update existing attributes with the datasetId	
Parent Release	v1.3.1	
Description	Check that one can update entity attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and	SUT <- Client

	<p>Query Parameter: fragment_filename set to 'vehicle-speed-two-datasetid-02-fragment.jsonld' and Query Parameter: content_type set to 'application/ld+json'</p>	
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 204 and Update Entity Attributes with Check Response Body Content and Query Parameter: expectation_filename set to 'expectation_resp_filename' and Query Parameter: response_body set to 'response.json()' and Query Parameter: additional_ignored_path set to '\$ {EMPTY}' and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_01_03	
Name	011_01_03 Check that one can update only some attributes while unknown are ignored	
Parent Release	v1.3.1	
Description	Check that one can update entity attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- speed-two-datasetid-03-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 204 and Update Entity Attributes with Check Response Body Content and Query Parameter: expectation_filename set to 'expectation_resp_filename' and Query Parameter: response_body set to 'response.json()' and Query Parameter: additional_ignored_path set to '\$ {EMPTY}' and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/011_03
Test objective	Check that one cannot update entity attributes if the entity id or attributes are not known to the system
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.2
Config Id	CF_01
PICS Selection	PICS_UpdAttEnt
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/011_03_01
Name	011_03_01 Update entity attributes when the entity id is not known to the system
Parent Release	v1.3.1

Description	Check that one cannot update entity attributes if the entity id or attributes are not known to the system	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/entities/{entityId}/attrs/{attributeId}' method set to 'PATCH' Update Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- speed-two-datasetid-01-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Update Entity Attributes with Response Status Code set to 404 and Update Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ ResourceNotFound' and Update Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client

4.4.3.3.4 AppendEntityAttributes

TP Id	TP/NGSI-LD/CI/Prov/EA/010_04	
Test objective	Check that one can append entity attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_04_01	
Name	010_04_01 Append entity attributes and ignore existing multi-attribute instance	
Parent Release	v1.3.1	
Description	Check that one can append entity attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/entities/{id}/attrs?options=\${options}' method set to 'POST' Append entity attributes with parameters: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- attribute-to-add-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' and Query Parameter: options set to 'noOverwrite' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes With Parameters with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and } </pre>	SUT -> Client

	Query Parameter: 'ignored_keys' set to 'ignored_attributes' }	
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_04_02	
Name	010_04_02 Append entity attributes with a new multi-attribute instance	
Parent Release	v1.3.1	
Description	Check that one can append entity attributes	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{id}/attrs?options=\${options}' method set to 'POST' Append entity attributes with parameters: and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- speed-different-datasetid-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' and Query Parameter: options set to 'noOverwrite' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Append Entity Attributes With Parameters with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/010_06	
Test objective	Check that one can append a LanguageProperty property to an entity	
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.5.18 and 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_LANG and PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_06_01	
Name	010_06_01 Append a LanguageProperty property	
Parent Release	v1.4.1	
Description	Check that one can append a LanguageProperty property to an entity	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/{id}/attrs' method set to 'POST' Append Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- new-language-property-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$	SUT -> Client

	{None}' }	
--	--------------	--

TP Id	TP/NGSI-LD/CI/Prov/EA/010_02	
Test objective	Check that one cannot append entity attributes with invalid/missing id or invalid request body	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_02_01	
Name	010_02_01 Append entity attributes if the entity Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot append entity attributes with invalid/missing id or invalid request body	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}/attrs' method set to 'POST' Append Entity Attributes and Query Parameter: id set to '\$ {EMPTY}' and Query Parameter: fragment_filename set to 'vehicle- attribute-to-add-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 400 and Append Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Append Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_02_02	
Name	010_02_02 Append entity attributes if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot append entity attributes with invalid/missing id or invalid request body	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}/attrs' method set to 'POST' Append Entity Attributes and Query Parameter: id set to 'thisisaninvaliduri' and Query Parameter: fragment_filename set to 'vehicle- attribute-to-add-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 400 and Append Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Append Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/010_07	
Test objective	Check that one can append a scope to an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clauses 4.18 and 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_SCOPES and PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_07_01	
Name	010_07_01 Append scope to an entity with overwrite enabled	
Parent Release	v1.5.1	
Description	Check that scope is replaced if overwrite is enabled	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/' method set to 'GET' Request Retrieve Entity by Id and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 204 and Append Entity Attributes with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_07_01	
Name	010_07_01 Append scope to an entity with overwrite disabled	
Parent Release	v1.5.1	
Description	Check that scope is appended if overwrite is disabled	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/' method set to 'GET' Request Retrieve Entity by Id and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: accept set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes With Parameters with Response Status Code set to 204 and Append Entity Attributes With Parameters with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/010_05	
Test objective	Check that one cannot append entity attributes with invalid/missing id or invalid request body	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_05_01	
Name	010_05_01 Append entity attributes with invalid entity fragments	
Parent Release	v1.3.1	
Description	Check that one cannot append entity attributes with invalid entity fragments	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-lid/v1/entities/{id}/attrs' method set to 'POST' Append Entity Attributes and Query Parameter: id set to '\${entity_id}' and Query Parameter: fragment_filename set to 'invalid-fragment.jsonld' and Query Parameter: content_type set to 'application/ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 400 and Append Entity Attributes with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ InvalidRequest' and Append Entity Attributes with Response body containing 'title' element } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/EA/010_01	
Test objective	Check that one can append entity attributes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.3	
Config Id	CF_01	
PICS Selection	PICS_AppAttEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Entity \${entity} with an id set to \${entityId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/EA/010_01_01	
Name	010_01_01 Append entity attributes	
Parent Release	v1.3.1	
Description	Check that one can append entity attributes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ ngsi-lid/v1/entities/{id}/attrs' method set to 'POST' Append Entity Attributes and Query Parameter: id set to '\$ {entity_id}' and Query Parameter: fragment_filename set to 'vehicle- new-attribute-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Append Entity Attributes with Response Status Code set to 204 and Retrieve Entity by Id with Check Updated Entity and Query Parameter: 'updated_resource' set to 'entity_expectation_payload' and </pre>	SUT -> Client

	ResourceNotFound' and Append Entity Attributes with Response body containing 'title' element }	
--	--	--

4.4.3.4 Entities

4.4.3.4.1 CreateEntity

TP Id	TP/NGSI-LD/CI/Prov/E/001_07	
Test objective	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_07_01	
Name	001_07_01 Create one entity using a JSON-LD @context obtained from the request payload with ld context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and retrieve with ld context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'building-simple-attributes-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_07_02	
Name	001_07_02 Create one entity using a JSON-LD @context obtained from the request payload without context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from the request payload body itself if the Content-Type header is "application/ld+json" and retrieve without context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'building-simple-attributes-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'https:// ngsi-ld-test-suite/context#almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_05	
Test objective	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_05_01	
Name	001_05_01 Create one entity using the default context with JSON content type and request without context	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context, requesting without context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload defined in file: 'building-simple-attributes-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_05_02	
Name	001_05_02 Create one entity using the default context with JSON content type and request with context	
Parent Release	v1.3.1	
Description	Check that the default @context is used if the Content-Type header is "application/json" and the Link header does not contain a JSON-LD @context, requesting with context	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload defined in file: 'building-simple-attributes-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'ngsi-ld:default-context/almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_01
Test objective	Check that one can create an entity
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.1
Config Id	CF_01
PICS Selection	PICS_EntCre
Initial conditions	with { the SUT containing an initial state }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_01_01

Name	001_01_01 MinimalEntity	
Parent Release	v1.3.1	
Description	Check that one can create an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/json' and payload defined in file: 'building-minimal- sample.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'created_entity' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_01_02	
Name	001_01_02 EntityWithSimpleProperties	
Parent Release	v1.3.1	
Description	Check that one can create an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-simple- attributes-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'created_entity' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_01_03	
Name	001_01_03 EntityWithRelationshipsProperties	
Parent Release	v1.3.1	
Description	Check that one can create an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- </pre>	SUT <- Client

	Type] set to 'application/ld+json' and payload defined in file: 'building-relationship-of-property-sample.jsonld'	
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'created_entity' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_01_04	
Name	001_01_04 EntityWithLocationAttribute	
Parent Release	v1.3.1	
Description	Check that one can create an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-location- attribute-sample.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'created_entity' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_03	
Test objective	Check that one cannot create an entity with an existing id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.1	
Config Id	CF_01	
PICS Selection	PICS_EntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_03_01	
Name	001_03_01 Create one valid entity and one invalid entity	
Parent Release	v1.3.1	
Description	Check that one cannot create an entity with an existing id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- </pre>	SUT <- Client

	Type'] set to 'application/ld+json' and payload defined in file: 'building-minimal-sample.jsonld' }	
	then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Status Code set to 409 and Create Entity Selecting Content Type with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/AlreadyExists' and Create Entity Selecting Content Type with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_04	
Test objective	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_04_01	
Name	001_04_01 Create one entity using a provided Link header with JSON content type and NGSILD context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload defined in file: 'building-simple-attributes-sample.json' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_04_02	
Name	001_04_02 Create one entity using a provided Link header with JSON content type and no context	
Parent Release	v1.3.1	
Description	Check that the @context is obtained from a Link Header if the Content-Type header is "application/json"	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and	SUT <- Client

	payload defined in file: 'building-simple-attributes-sample.json' }	
	then { the SUT sends a valid Response for the operation: Retrieve Entity by Id with Check Response Body containing an Attribute set to and Query Parameter: expected_attribute_name set to 'https://ngsi-ld-test-suite/context#almostFull' and Query Parameter: response_body set to 'response1.json()' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_06	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_06_01	
Name	001_06_01 Create one entity containing a JSON-LD @context with a JSON content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/json" and the request payload body (as JSON) contains a "@context" term	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/json' and payload defined in file: 'building-simple-attributes-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 400 and Create Entity Selecting Content Type with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/BadRequestData' and Create Entity Selecting Content Type with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_02	
Test objective	Check that one cannot create an entity with an invalid request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.1	
Config Id	CF_01	
PICS Selection	PICS_EntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_02_01	
Name	001_02_01 InvalidJson	
Parent Release	v1.3.1	
Description	Check that one cannot create an entity with an invalid request	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request creation of an entity from filename 'invalid- json-sample.jsonld' and Content- Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity From File with Response Status Code set to 400 and Create Entity From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Create Entity From File with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_02_02	
Name	001_02_02 EmptyJson	
Parent Release	v1.3.1	
Description	Check that one cannot create an entity with an invalid request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request creation of an entity from filename 'empty- sample.jsonld' and Content- Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity From File with Response Status Code set to 400 and Create Entity From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ InvalidRequest' and Create Entity From File with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_02_03	
Name	001_02_03 EntityWithNoContext	
Parent Release	v1.3.1	
Description	Check that one cannot create an entity with an invalid request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request creation of an entity from filename 'building- minimal-sample.json' and Content- Type set to 'application/ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity From File with Response Status Code set to 400 and Create Entity From File with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Entity From File with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_11	
Test objective	Check that one can create an entity with one or more scopes	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 4.18	
Config Id	CF_01	
PICS Selection	PICS_SCOPES	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_11_01	
Name	001_11_01 EntityWithOneScope	
Parent Release	v1.5.1	
Description	Check that one can create an entity with one or more scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content- Type'] set to 'application/json' and payload defined in file: 'building-minimal- with-one-scope.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_11_02	
Name	001_11_02 EntityWithManyScopes	
Parent Release	v1.5.1	
Description	Check that one can create an entity with one or more scopes	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/ method set to 'POST' Request Header['Content- Type'] set to 'application/json' and payload defined in file: 'building-minimal- with-many-scopes.json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Create Entity Selecting Content Type with Response Header: Location containing \${entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/001_08	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_08_01	
Name	001_08_01 Create one entity not containing a JSON-LD @context with a JSON-LD content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and the request payload body does not contain a @context term	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-simple- attributes-sample.json' } then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 400 and Create Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Entity Selecting Content Type with Response body containing 'title' element } </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

TP Id	TP/NGSI-LD/CI/Prov/E/001_10	
Test objective	Check that one can create an entity with a LanguageProperty property	
Reference	ETSI GS CIM 009 V1.4.1 [1], clauses 4.5.18 and 5.6.1	
Config Id	CF_01	
PICS Selection	PICS_LANG and PICS_EntCre	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_10_01	
Name	001_10_01 EntityWithLanguageProperty	
Parent Release	v1.4.1	
Description	Check that one can create an entity with a LanguageProperty property	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-language- property-sample.jsonld' } then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

	201 and Create Entity Selecting Content Type with Response Header: Location containing \$\$entity_id} and Retrieve Entity by Id with Check Created Resource Set To and Query Parameter: 'created_resource' set to 'created_entity' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to '\$ {None}' list of keys }	
--	---	--

TP Id	TP/NGSI-LD/CI/Prov/E/001_09	
Test objective	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 6.3.5	
Config Id	CF_01	
PICS Selection	PICS_HTTP_CONTEXT_RESOLV	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/001_09_01	
Name	001_09_01 Create one entity with a Link header and a JSON-LD content type	
Parent Release	v1.3.1	
Description	Check that an HTTP error response of type BadRequestData is raised if the Content-Type header is "application/ld+json" and a JSON-LD Link header is present in the incoming HTTP request	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content-Type'] set to 'application/ld+json' and payload defined in file: 'building-simple-attributes-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 400 and Create Entity Selecting Content Type with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Create Entity Selecting Content Type with Response body containing 'title' element }	SUT -> Client

4.4.3.4.2 DeleteEntity

TP Id	TP/NGSI-LD/CI/Prov/E/002_03	
Test objective	Check that one cannot delete an entity if the entity id is not known to the system	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.6	
Config Id	CF_01	
PICS Selection	PICS_EntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/002_03_01	
Name	002_03_01 Delete an entity with an id not known to the system	
Parent Release	v1.3.1	
Description	Check that one cannot delete an entity if the entity id is not known to the system	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'DELETE' Delete Entity Request with id set to '\$ {entity_id}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Entity by Id with Response Status Code set to 404 and Delete Entity by Id with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Delete Entity by Id with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/002_01	
Test objective	Check that one can delete an entity by id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.6	
Config Id	CF_01	
PICS Selection	PICS_EntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/002_01_01	
Name	002_01_01 Delete an entity	
Parent Release	v1.3.1	
Description	Check that one can delete an entity by id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/' method set to 'POST' Request Header['Content- Type'] set to 'application/ld+json' and payload defined in file: 'building-simple- attributes-sample.jsonld' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Entity Selecting Content Type with Response Status Code set to 201 and Delete Entity by Id with Response Status Code set to 204 and Retrieve Entity by Id with Response Status Code set to \${response2.status_code} }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/E/002_02	
Test objective	Check that one cannot delete an entity with invalid/missing id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.6	
Config Id	CF_01	
PICS Selection	PICS_EntDel	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/E/002_02_01	
Name	002_02_01 Delete an entity if the Entity Id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot delete an entity with invalid/missing id	

Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'DELETE' Delete Entity Request with id set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Entity by Id with Response Status Code set to 405 and Delete Entity by Id with Response Body containing the type '\$ {EMPTY}' and Delete Entity by Id with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/E/002_02_02	
Name	002_02_02 Delete an entity if the Entity Id is not a valid URI	
Parent Release	v1.3.1	
Description	Check that one cannot delete an entity with invalid/missing id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/entities/{id}' method set to 'DELETE' Delete Entity Request with id set to 'thisisaninvaliduri' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Delete Entity by Id with Response Status Code set to 400 and Delete Entity by Id with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ BadRequestData' and Delete Entity by Id with Response body containing 'title' element }	SUT -> Client

4.4.3.5 TemporalEntityAttributes

4.4.3.5.1 DeleteAttribute

TP Id	TP/NGSI-LD/CI/Prov/TEA/015_03	
Test objective	Check that an error is raised if one deletes a temporal entity with an unknown EntityId/Attribute Id	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.13	
Config Id	CF_01	
PICS Selection	PICS_AttTemEvoEntDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_03_01	
Name	015_03_01 Delete an attribute to a temporal entity if the entity id does not exist	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes a temporal entity with an unknown EntityId/Attribute Id	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{entityId}/attrs/{attributeId}?\${ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and	SUT <- Client

	<p>Query Parameter: entityId set to '\$ {unknown_temporal_entity_id}' and Query Parameter: attributeId set to 'fuelLevel' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</p>	
	<p>then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 404 and Delete Attribute From Temporal Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Delete Attribute From Temporal Entity with Response body containing 'title' element }</p>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_03_02	
Name	015_03_02 Delete an attribute to a temporal entity if the entity does not contain the target attribute	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes a temporal entity with an unknown EntityId/Attribute Id	
Expected behaviour	Test events	Direction
	<p>when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {valid_temporal_entity_id}' and Query Parameter: attributeId set to 'notExistingAttribute' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' }</p>	SUT <- Client
	<p>then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 404 and Delete Attribute From Temporal Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-ld/errors/ ResourceNotFound' and Delete Attribute From Temporal Entity with Response body containing 'title' element }</p>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/015_01
Test objective	Check that one can delete an attribute of a temporal representation of an entity with simple temporal properties
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.13

Config Id	CF_01	
PICS Selection	PICS_AttTemEvoEntDel	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \$ {entity} with an id set to \$ {temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_01_01	
Name	015_01_01 Delete an attribute from a temporal representation of an entity without deleteAll/datasetId	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute of a temporal representation of an entity with simple temporal properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to 'fuelLevel' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {datasetid}' and Query Parameter: deleteAll set to '\$ {deleteall}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'updated_resource' set to 'temporal_entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_01_02	
Name	015_01_02 Delete an attribute from a temporal representation of an entity with datasetId	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute of a temporal representation of an entity with simple temporal properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to 'fuelLevel' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {datasetid}' and Query Parameter: deleteAll set to '\$ {deleteall}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ </pre>	SUT <- Client

	<pre> jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'updated_resource' set to 'temporal_entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_01_03	
Name	015_01_03 Delete an attribute from a temporal representation of an entity with deleteAll	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute of a temporal representation of an entity with simple temporal properties	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to 'fuelLevel' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {datasetid}' and Query Parameter: deleteAll set to '\$ {deleteall}' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'updated_resource' set to 'temporal_entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/015_02
Test objective	Check that an error is raised if one deletes an attribute to temporal entity with an unknown/invalid Entity/Attribute Id
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.13
Config Id	CF_01
PICS Selection	PICS_AttTemEvoEntDel
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${attrId} }
Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_02_01
Name	015_02_01 Delete an attribute to a temporal representation of an entity with a missing entity id
Parent Release	v1.3.1

Description	Check that an error is raised if one deletes an attribute to temporal entity with an unknown/invalid Entity/Attribute Id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {EMPTY}' and Query Parameter: attributeId set to 'speed' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 400 and Delete Attribute From Temporal Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Delete Attribute From Temporal Entity with Response body containing 'title' element } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_02_02	
Name	015_02_02 Delete an attribute to a temporal representation of an entity with an invalid entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes an attribute to temporal entity with an unknown/invalid Entity/Attribute Id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-lid/v1/temporal/entities/{entityId}/attrs/{attributeId}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to 'invalidId' and Query Parameter: attributeId set to 'speed' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 400 and Delete Attribute From Temporal Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-lid/errors/ BadRequestData' and Delete Attribute From Temporal Entity with Response body containing </pre>	SUT -> Client

	'title' element }	
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/015_02_03	
Name	015_02_03 Delete an attribute to a temporal representation of an entity with an invalid attribute id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one deletes an attribute to temporal entity with an unknown/invalid Entity/Attribute Id	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/temporal/entities/{entityId}/attrs/{attributeld}?\$ {params_as_string}' method set to 'DELETE' Delete Attribute From Temporal Entity: and Query Parameter: entityId set to '\$ {valid_temporal_entity_id}' and Query Parameter: attributeld set to 'invalid(Name' and Query Parameter: content_type set to 'application/ json' and Query Parameter: datasetId set to '\$ {EMPTY}' and Query Parameter: deleteAll set to 'false' and Query Parameter: context set to '\$ {EMPTY}' } then { the SUT sends a valid Response for the operations: Delete Attribute From Temporal Entity with Response Status Code set to 400 and Delete Attribute From Temporal Entity with Response Body containing the type 'https:// uri.etsi.org/ngsi-Id/errors/ BadRequestData' and Delete Attribute From Temporal Entity with Response body containing 'title' element } </pre>	<p>SUT <- Client</p> <p>SUT -> Client</p>

4.4.3.5.2 DeleteAttributeInstance

TP Id	TP/NGSI-LD/CI/Prov/TEA/017_01	
Test objective	Check that one can delete an attribute instance in temporal representation of an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.15	
Config Id	CF_01	
PICS Selection	PICS_AttnsTemEvoEnt	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_01_01	
Name	017_01_01 Delete an attribute instance in temporal representation of an entity	
Parent Release	v1.3.1	
Description	Check that one can delete an attribute instance in temporal representation of an entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-Id/v1/temporal/entities/{temporal_entity_id}/attrs/speed/ {instanceld}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeld set to 'speed' and Query Parameter: instanceld set to '\$ {instanceld}' and </pre>	SUT <- Client

	<p>Query Parameter: content_type set to 'application/json' and</p> <p>Query Parameter: context set to 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/jsonld-contexts/ngsi-ld-test-suite-compound.jsonld'</p>	
	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Delete Attribute Instance From Temporal Entity with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'updated_resource' set to 'temporal_entity_expectation_payload' and Query Parameter: 'response_body' set to 'response1.json()' and Query Parameter: 'ignored_keys' set to 'ignored_attributes' } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/017_03	
Test objective	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not right	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.15	
Config Id	CF_01	
PICS Selection	PICS_AttrInsTemEvoEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_03_01	
Name	017_03_01 Delete an attribute instance in temporal representation of an entity if the entity with given id is not found	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not found	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/{instancelId}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\${unknown_temporal_entity_id}' and Query Parameter: attributeId set to '\${attributeId}' and Query Parameter: instancelId set to '\${instancelId}' and Query Parameter: content_type set to 'application/json' and Query Parameter: context set to 'https://forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_03_02	
Name	017_03_02 Delete an attribute instance in temporal representation of an entity if the target attribute is not found	
Parent Release	v1.3.1	

Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not found	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instancelId}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instancelId set to '\$ {instancelId}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_03_03	
Name	017_03_03 Delete an attribute instance in temporal representation of an entity if the target attribute instance is not found	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not found	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instancelId}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instancelId set to '\$ {instancelId}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/017_02
Test objective	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not right
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.15
Config Id	CF_01
PICS Selection	PICS_AttnsTemEvoEnt

Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} and an attribute with an id set to \${attrId} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_02_01	
Name	017_02_01 Delete an attribute instance in temporal representation of an entity if the entity id is not valid	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to 'invalidld' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_02_02	
Name	017_02_02 Delete an attribute instance in temporal representation of an entity if the entity id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {EMPTY}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_02_03	

Name	017_02_03 Delete an attribute instance in temporal representation of an entity if the instance id is not valid	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instancelId} method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instancelId set to '\$ {instancelId}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_02_04	
Name	017_02_04 Delete an attribute instance in temporal representation of an entity if the attribute name is not a valid name	
Parent Release	v1.3.1	
Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/InstancelId is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instancelId} method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instancelId set to '\$ {instancelId}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/017_02_05	
Name	017_02_05 Delete an attribute instance in temporal representation of an entity if the attribute name is not present	
Parent Release	v1.3.1	

Description	Check that one cannot delete an attribute instance in temporal representation of an entity if the EntityId/AttributeId/Instanceid is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instanceid}' method set to 'DELETE' Delete Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instanceid set to '\$ {instanceid}' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to 'https:// forge.etsi.org/rep/cim/ngsi-ld-test-suite/-/raw/develop/resources/ jsonld-contexts/ngsi-ld-test-suite-compound.jsonld' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Delete Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client

4.4.3.5.3 AddAttributes

TP Id	TP/NGSI-LD/CI/Prov/TEA/014_03	
Test objective	Check that an error is raised if one adds an attribute to a non-existent entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.12	
Config Id	CF_01	
PICS Selection	PICS_TemEvoAttEntIns	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/014_03_01	
Name	014_03_01 Add Attribute To Temporal Entity	
Parent Release	v1.3.1	
Description	Check that an error is raised if one adds an attribute to a non-existent entity	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs' method set to 'POST' Append Attribute to Temporal Entity and Query Parameter: id set to '\$ {not_found_temporal_entity_representation_id}' and Query Parameter: fragment_filename set to 'vehicle- temporal-representation-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Append Attribute To Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client

	Id+json' }	
	then { the SUT sends a valid Response for the operations: Append Attribute To Temporal Entity with Response Status Code set to 400 and Append Attribute To Temporal Entity with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/InvalidRequest' and Append Attribute To Temporal Entity with Response body containing 'title' element }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/014_04_02	
Name	014_04_02 Add an attribute to a temporal representation of an entity with empty content	
Parent Release	v1.3.1	
Description	Check that an error is raised if one adds a temporal entity attribute with empty/invalid content	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs' method set to 'POST' Append Attribute to Temporal Entity and Query Parameter: id set to '\$ {temporal_entity_representation_id}' and Query Parameter: fragment_filename set to 'vehicle-temporal-representation-empty-json-fragment.jsonld' and Query Parameter: content_type set to 'application/Id+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Append Attribute To Temporal Entity with Response Status Code set to 400 and Append Attribute To Temporal Entity with Response Body containing the type 'https://uri.etsi.org/ngsi-ld/errors/InvalidRequest' and Append Attribute To Temporal Entity with Response body containing 'title' element }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/014_02	
Test objective	Check that an error is raised if one adds an attribute to a temporal entity with invalid content	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.12	
Config Id	CF_01	
PICS Selection	PICS_TemEvoAttEntIns	
Initial conditions	with { the SUT containing an initial state }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/014_02_01	
Name	014_02_01 Add an attribute to a temporal representation of an entity with an empty entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one adds a temporal entity attribute with a non-existing/invalid EntityId	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs' method set to 'POST' Append Attribute to Temporal Entity and Query Parameter: id set to '\$ {EMPTY}' and Query Parameter: fragment_filename set to 'vehicle-	SUT <- Client

	temporal-representation-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	
	then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Append Attribute To Temporal Entity with Response Status Code set to 400 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/014_02_02	
Name	014_02_02 Add an attribute to a temporal representation of an entity with an invalid entity id	
Parent Release	v1.3.1	
Description	Check that an error is raised if one adds a temporal entity attribute with a non-existing/invalid EntityId	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs' method set to 'POST' Append Attribute to Temporal Entity and Query Parameter: id set to 'thisIsAnInvalidId' and Query Parameter: fragment_filename set to 'vehicle- temporal-representation-fragment.jsonld' and Query Parameter: content_type set to 'application/ ld+json' }	SUT <- Client
	then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Append Attribute To Temporal Entity with Response Status Code set to 400 }	SUT -> Client

4.4.3.5.4 PartialUpdateAttributeInstance

TP Id	TP/NGSI-LD/CI/Prov/TEA/016_01	
Test objective	Check that one can modify an attribute instance in temporal representation of an entity	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.14	
Config Id	CF_01	
PICS Selection	PICS_AttrInsTemEvoEnt	
Initial conditions	with { the SUT containing an initial state }	
	Permutations	
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_01_01	
Name	016_01_01 Modify attribute instance in temporal representation of an entity	
Parent Release	v1.3.1	
Description	Check that one can partially update an attribute instance of a temporal representation of an entity	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to '/ ngsi-ld/v1/temporal/entities' method set to 'POST' Create or Update Temporal Representation of Entity Selecting Content Type: and Query Parameter: temporal_entity_representation_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: filename set to 'vehicle- temporal-representation-sample.jsonld' and Query Parameter: content_type set to 'application/'	SUT <- Client

	Id+json' and Query Parameter: accept set to '\$ {EMPTY}' }	
	then { the SUT sends a valid Response for the operations: Create Or Update Temporal Representation Of Entity Selecting Content Type with Response Status Code set to 201 and Modify Attribute Instance From Temporal Entity with Response Status Code set to 204 and Retrieve Temporal Representation Of Entity with Check Updated Entity and Query Parameter: 'ignored_keys' set to '\$ {None}' }	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/016_03	
Test objective	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelid is not found	
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.14	
Config Id	CF_01	
PICS Selection	PICS_AttrInsTemEvoEnt	
Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_03_01	
Name	016_03_01 Modify attribute instance in temporal representation of an entity if the entity with given id is not found	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelid is not found	
Expected behaviour	Test events	Direction
	when { the SUT receives a Request from the client containing: URL set to / ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelid}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {unknown_temporal_entity_id}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelid set to '\$ {instancelid}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' }	SUT <- Client
	then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 404 }	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_03_02	
Name	016_03_02 Modify attribute instance in temporal representation of an entity if the target attribute is not found	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelid is not found	

Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instanceld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instanceld set to '\$ {instanceld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_03_03	
Name	016_03_03 Modify attribute instance in temporal representation of an entity if the target attribute instance is not found	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/AttributeId/Instanceld is not found	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeId}/ {instanceld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeId set to '\$ {attributeId}' and Query Parameter: instanceld set to '\$ {instanceld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 404 } </pre>	SUT -> Client

TP Id	TP/NGSI-LD/CI/Prov/TEA/016_02
Test objective	Check that one cannot modify an attribute instance in temporal representation of an entity if the EntityId/AttributeId/Instanceld is not right
Reference	ETSI GS CIM 009 V1.5.1 [1], clause 5.6.14
Config Id	CF_01
PICS Selection	PICS_AttnsTemEvoEnt

Initial conditions	with { the SUT being in the "initial state" and the SUT containing an initial Temporal Entity \${entity} with an id set to \${temporal_entity_representation_id} }	
Permutations		
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_01	
Name	016_02_01 Modify attribute instance in temporal representation of an entity if the entity id is not valid	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to 'invalidld' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_02	
Name	016_02_02 Modify attribute instance in temporal representation of an entity if the entity id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {EMPTY}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client

TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_03	
Name	016_02_03 Modify attribute instance in temporal representation of an entity if the instance id is not valid	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_04	
Name	016_02_04 Modify attribute instance in temporal representation of an entity if the instance id is not present	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instancelld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instancelld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instancelld set to '\$ {instancelld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 405 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_05	

Name	016_02_05 Modify attribute instance in temporal representation of an entity if the attribute name is not a valid name	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instanceld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instanceld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instanceld set to '\$ {instanceld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 400 } </pre>	SUT -> Client
TP Permutation id	TP/NGSI-LD/CI/Prov/TEA/016_02_06	
Name	016_02_06 Modify attribute instance in temporal representation of an entity if the attribute name is not present	
Parent Release	v1.3.1	
Description	Check that one cannot partially modify attribute instance in temporal representation of an entity if the EntityId/Attributeld/Instanceld is not right	
Expected behaviour	Test events	Direction
	<pre> when { the SUT receives a Request from the client containing: URL set to '/' ngsi-ld/v1/temporal/entities/{temporal_entity_id}/attrs/{attributeld}/ {instanceld}' method set to 'PATCH' Modify Attribute Instance from Temporal Entity: and Query Parameter: temporal_entity_id set to '\$ {temporal_entity_representation_id}' and Query Parameter: attributeld set to '\$ {attributeld}' and Query Parameter: instanceld set to '\$ {instanceld}' and Query Parameter: fragment_filename set to 'vehicle- temporal-modify-attribute-instance-fragment.jsonld' and Query Parameter: content_type set to 'application/ json' and Query Parameter: context set to '\$ {EMPTY}' } </pre>	SUT <- Client
	<pre> then { the SUT sends a valid Response for the operation: Modify Attribute Instance From Temporal Entity with Response Status Code set to 405 } </pre>	SUT -> Client

Annex A (informative): Bibliography

- ISO 8601:2004: "Data elements and interchange formats -- Information interchange --Representation of dates and times".

Annex B (informative): Change History

Date	Version	Information about changes
October, 19 th 2020	V0.0.1	First draft of document
February, 4 th 2021	V1.0.1	Stable draft approved by ISG-CIM
March, 23 rd 2021	V1.0.2	
April 28 th 2021	V1.1.1	Last Technical Officer review for Publication pre-processing with ETSI EditHelp
October, 4 th 2023	V1.2.1	Stable draft approved by ISG-CIM
November 2023	V1.3.1	First draft version of automatically generated content
December 2023	V1.4.1	Second draft version of automatically generated content
February 2024	V1.5.1	Third draft version of automatically generated content
March 2024	V1.6.1	Fourth draft version of automatically generated content. Cleanup revisions
March 2024	V1.7.1	Fifth draft version of automatically generated content, including v1.4 and v1.5 tests. Technical Officer review for Publication pre-processing with ETSI EditHelp

History

Document history		
V1.1.1	May 2021	Publication
V2.1.1	April 2024	Publication