



GROUP SPECIFICATION

**Network Functions Virtualisation (NFV) Release 6;
Protocols and Data Models;
Specification of protocol and data model solutions
for PIM services**

Reference

DGS/NFV-SOL026

Keywords

data models, MANO, NFV, PIM

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 the
[ETSI Search & Browse Standards](#) application.

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 on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

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 of this document may be reproduced in any form, by any means and in any media, without the prior written authorization of ETSI and except as expressly permitted below.

By way of exception and when the document is a normative deliverable (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)), ETSI authorizes to reproduce and incorporate into products, services and technical documentation only those extracts (e.g. templates) that are strictly necessary for the technical implementation of the normative deliverable, to ensure compliance with the latter. Nothing in this notice shall be construed as limiting any mandatory exceptions to copyright provided by applicable law.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights	7
Foreword.....	7
Modal verbs terminology.....	7
1 Scope	8
2 References	8
2.1 Normative references	8
2.2 Informative references.....	8
3 Definition of terms, symbols and abbreviations.....	9
3.1 Terms.....	9
3.2 Symbols.....	9
3.3 Abbreviations	9
4 Overview	9
4.1 Summary of ETSI GS NFV-IFA 053.....	9
4.2 Summary of ETSI GS NFV-IFA 010.....	10
4.3 Profiled protocol and data model solution: Redfish® interface	10
4.3.1 Introduction.....	10
4.3.2 API structure	11
4.3.3 Data Model Concepts.....	11
5 Mapping of NFV object model to profiled solution objects.....	12
5.1 Overview	12
5.2 Common data objects	12
5.2.1 Subscription	12
5.2.2 Notification	12
5.2.3 Task	13
5.3 Mapping of PIM information model with Redfish® objects.....	13
5.3.1 Physical resource object.....	13
5.3.2 Resource zone object	13
5.3.3 Physical resource pool object.....	13
5.3.4 NFVI-PoP object	14
5.4 Performance management related objects	15
5.4.1 PMJob	15
5.4.2 Threshold	15
5.4.3 PerformanceReport	15
5.5 Log management related objects	15
5.5.1 LoggingJob	15
5.5.2 Log.....	16
5.6 Fault management related objects	16
5.6.1 Alarm	16
6 Physical resource provisioning and lifecycle management service interface.....	16
6.1 Description	16
6.2 API version.....	17
6.3 Resource structure and HTTP methods	17
6.4 Sequence diagrams (informative).....	18
6.4.1 Introduction.....	18
6.4.2 Creation of a Computer System resource	18
6.4.3 Querying information about Computer System resources	19
6.4.4 Modifying a Computer System resource	20
6.4.5 Replacing a Computer System resource	21
6.4.6 Deleting a Computer System resource.....	22
6.4.7 Perform an action on a Computer System resource	23
6.5 Resources	25
6.5.1 Introduction.....	25
6.5.2 Resource: Systems	25

6.5.3	Resource: Individual Computer System	25
6.5.4	Resource: Individual Computer System Action.....	26
6.5.5	Resource: Subscriptions.....	26
6.5.6	Resource: Individual Subscription	26
6.5.7	Resource: Individual Task Monitor	27
6.6	Data model	27
7	Physical resource inventory management service interface	27
7.1	Description	27
7.2	API version.....	28
7.3	Resource structure and HTTP methods	28
7.4	Sequence diagrams (informative).....	29
7.4.1	Introduction.....	29
7.4.2	Querying information about inventory objects	30
7.4.3	Modifying an inventory object.....	30
7.4.4	Replacing an inventory object	31
7.5	Resources	32
7.5.1	Introduction.....	32
7.5.2	Resource: InventoryObjectCollection.....	32
7.5.3	Resource: IndividualInventoryObject	33
7.5.4	Resource: Subscriptions.....	33
7.5.5	Resource: Individual Subscription	34
7.5.6	Resource: Individual Task Monitor	34
7.6	Data model	34
8	Physical resource topology management service interface	35
8.1	Description	35
8.2	API version.....	35
8.3	Resource structure and HTTP methods	35
8.4	Sequence diagrams (informative).....	37
8.4.1	Introduction.....	37
8.4.2	Creation of a topology object.....	37
8.4.3	Querying information about topology objects	38
8.4.4	Modifying a topology object.....	39
8.4.5	Replacing a topology object.....	39
8.4.6	Deleting a topology object	40
8.5	Resources	41
8.5.1	Introduction.....	41
8.5.2	Resource: Resource Zones.....	41
8.5.3	Resource: Individual Resource Zone	42
8.5.4	Resource: Resource Blocks.....	42
8.5.5	Resource: Individual Resource Block.....	43
8.5.6	Resource: Subscriptions.....	43
8.5.7	Resource: Individual Subscription	44
8.5.8	Resource: Individual Task Monitor	44
8.6	Data model	44
9	Physical resource performance management service interface	45
9.1	Description	45
9.2	API version.....	45
9.3	Resource structure and HTTP methods	45
9.4	Sequence diagrams (informative).....	47
9.4.1	Introduction.....	47
9.4.2	Creation of a PM object	47
9.4.3	Querying information about PM objects	48
9.4.4	Modifying a PM object	49
9.4.5	Replacing a PM object	50
9.4.6	Deleting a PM object	51
9.4.7	Retrieving performance reports	52
9.4.8	Receiving performance reports asynchronously	53
9.5	Resources	54
9.5.1	Introduction.....	54
9.5.2	Resource: PmJobs	54

9.5.3	Resource: Individual PmJob	54
9.5.4	Resource: Thresholds.....	55
9.5.5	Resource: Individual Threshold.....	55
9.5.6	Resource: Performance Reports.....	56
9.5.7	Resource: Individual Performance Report.....	56
9.5.8	Resource: Subscriptions.....	56
9.5.9	Resource: Individual Subscription.....	57
9.5.10	Resource: Individual Task Monitor	57
9.6	Data model	58
10	Physical resource fault management service interface	58
10.1	Description	58
10.2	API version.....	58
10.3	Resource structure and HTTP methods	58
10.4	Sequence diagrams (informative).....	59
10.4.1	Introduction.....	59
10.5	Resources	60
10.5.1	Introduction.....	60
10.5.2	Resource: Subscriptions.....	60
10.5.3	Resource: Individual Subscription.....	60
10.5.4	Resource: Individual Task Monitor	61
10.6	Data model	61
11	Physical resource log management service interface	61
11.1	Description	61
11.2	API version.....	61
11.3	Resource structure and HTTP methods	62
11.4	Sequence diagrams (informative).....	63
11.4.1	Introduction.....	63
11.4.2	Creation of a Logging Job	63
11.4.3	Querying information about Logging Job resources.....	64
11.4.4	Modifying a Logging Job resource	65
11.4.5	Replacing a Logging Job resource	66
11.4.6	Deleting a Logging Job resource	67
11.4.7	Querying information about logs	68
11.4.8	Receiving diagnostics related logs	69
11.4.9	Receiving notification about availability of diagnostics related logs.....	70
11.5	Resources	71
11.5.1	Introduction.....	71
11.5.2	Resource: LoggingJobs.....	71
11.5.3	Resource: Individual LoggingJob.....	71
11.5.4	Resource: Logs	72
11.5.5	Resource: Individual Log.....	72
11.5.6	Resource: Subscriptions.....	73
11.5.7	Resource: Individual Subscription.....	73
11.5.8	Resource: Individual Task Monitor	73
11.6	Data model	74
Annex A (informative):	Sequence Diagrams.....	75
A.1	Overview	75
A.2	Flow of managing subscriptions.....	75
A.3	Flow of sending notifications	77
A.4	Flow of monitoring tasks related to asynchronous operations (polling method)	78
A.5	Flow of monitoring tasks related to asynchronous operations (subscribe/notify method).....	79
Annex B (informative):	Relationship between the PIM and the standard Redfish® service	81
B.1	Introduction	81
B.2	Management considerations for the distributed architecture.....	82

B.2.1	General considerations	82
B.2.2	Considerations for inventory management	82
Annex C (informative):	Change history	83
History		85

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)) 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 IPR online database](#).

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™**, **LTE™** and **5G™** logo 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) Network Functions Virtualisation (NFV).

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 specifies the protocol and data model to fulfil the requirements of Physical Infrastructure Management (PIM) as specified in ETSI GS NFV-IFA 053 [1]. It analyses the PIM information model and the managed objects of the referenced solutions like Redfish® and provides a mapping between them. Wherever applicable, the referenced solutions are profiled for specifying the physical resource provisioning and lifecycle management, inventory management, topology management, performance management, fault management and log management service interfaces.

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 in the [ETSI docbox](#).

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 NFV-IFA 053](#): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements and interface specification for Physical Infrastructure Management".
- [2] [DMTF® DSP0266 \(V1.23.1\)](#): "Redfish Specification".
- [3] [DMTF® DSP0268 \(V2025.4\)](#): "Redfish Data Model Specification".

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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

- [i.1] ETSI GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".
- [i.2] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Functional requirements specification".
- [i.3] ODIM Project: "[Open Distributed Infrastructure Management \(ODIM\)](#)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR NFV 003 [i.1] apply.

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in ETSI GR NFV 003 [i.1].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR NFV 003 [i.1] and the following apply:

CSDL	Common Schema Definition Language
JSON	JavaScript Object Notation
URI	Unique Resource Identifier

4 Overview

4.1 Summary of ETSI GS NFV-IFA 053

ETSI GS NFV-IFA 053 [1] specifies requirements related to physical infrastructure management services provided by the PIM function. Service interfaces related to following management aspects are produced by the PIM:

- 1) Physical resource provisioning and lifecycle management.
- 2) Physical resource inventory management.
- 3) Physical resource topology management.
- 4) Physical resource performance management.
- 5) Physical resource fault management.
- 6) Physical resource log management.

Provisioning and lifecycle management service interface produced by the PIM enables consumers to set up and provision physical resources, including composed physical resources that comprise of multiple physical resources. It also enables physical resource lifecycle management operations including creation, update, querying and deletion of corresponding managed objects related to physical resources that are managed by the PIM. Furthermore, this service interface supports sending notifications to the subscribed consumers about changes related to physical resources.

Inventory management service interface produced by the PIM enables consumers to query and update inventory information related to physical resources in the NFVI. The service interface also supports providing notifications to subscribed consumers regarding any changes in the inventory information.

Topology management service interface produced by the PIM enables consumers to query topology information related to physical resources, which includes the physical topology information and the logical topology information. Physical topology refers to the relationship between physical resources from physical perspective and logical topology refers to the relationship between physical resources from a logical perspective. These concepts are described in detail in clause 6 of ETSI GS NFV-IFA 053 [1]. Additionally, this service interface supports management of logical topology information including the creation, update and deletion of physical resource pools. It also supports providing notifications to subscribed consumers in case of any changes in topology information of physical resources.

Performance management interface produced by the PIM enables consumers to collect performance information related to physical resources. The interface also supports creation, update, query and deletion of PM jobs and thresholds so the consumers can control the collection and reporting of performance information.

Fault management service interface produced by the PIM supports providing alarm information related to physical resources as well as sending notifications regarding alarm information. The service interface also supports managing alarms, such as clearing or acknowledging alarms.

Log management service interface produced by the PIM enables consumers to query log information of physical resources. The service interface supports controlling the collection of log information as well as sending notifications regarding the availability of logs related to physical resources.

4.2 Summary of ETSI GS NFV-IFA 010

ETSI GS NFV-IFA 010 [i.2] specifies that the following statements on the scope of the PIM function, part of NFV-MANO, apply to all PIM related requirements:

- The PIM is responsible for provisioning and managing lifecycle, inventory, topology, log, configuration, performance and fault management of physical compute, storage, and networking resources in the NFVI.
- The PIM exposes corresponding interfaces to its consumers and translates incoming request into operations which are enforced towards the required actions to the managed physical resources.

The functional requirements on the PIM function in ETSI GS NFV-IFA 010 [i.2] are grouped into requirements per the following PIM management services and some additional non-service-related requirements regarding, e.g. connectivity management and energy management related to physical resources.

- provisioning and lifecycle management;
- inventory management;
- topology management;
- performance management;
- fault management; and
- log management.

According to ETSI GS NFV-IFA 010 [i.2], more detailed information about the PIM services and information model is provided in ETSI GS NFV-IFA 053 [1].

4.3 Profiled protocol and data model solution: Redfish® interface

4.3.1 Introduction

The present clause provides an overview of Redfish® interface specification [2], which is the profiled solution for the following physical resource management service interfaces produced by the PIM as specified in ETSI GS NFV-IFA 053 [1]:

- physical resource provisioning and lifecycle management service interface;
- physical resource inventory management service interface;
- physical resource topology management service interface;
- physical resource performance management service interface;
- physical resource fault management service interface; and

- physical resource log management service interface.

The overview consists of the high-level API structure of the Redfish® interface [2] and its supported data model for the protocol and data model specification of relevant PIM service interfaces, respectively.

4.3.2 API structure

Redfish® offers a RESTful interface for the clients, i.e. interface consumers, to interact with different Redfish® services. The Redfish® interface specification [2] covers the protocol details, RESTful API, its resources, resource identifiers, HTTP methods, request response schemas and other details like HTTP headers. Some important characteristics of the Redfish® RESTful interface are summarized below:

- Each resource is uniquely identified using a Unique Resource Identifier (URI) and can be accessed at its specific URI.
- Each resource is represented by its data model and is retrievable over the interface as a JSON payload.
- Each resource is strongly typed and is identified by the type identifier in the response body. The type identifier corresponds to the schema definition for the resource objects. Redfish® Data Model Specification [3] schema are available in JSON schema, OpenAPI and CSDL format.
- A set of resources with the same schema is called a resource collection.

The basic resource structure of Redfish® interface [2] can be generalized using the following patterns:

- /redfish/v1
- /redfish/v1/{resource}
- /redfish/v1/{resource}/{resource}
- /redfish/v1/{resource}/{resource}/.../{resource}; where {resource} can be {collection resource}/{resource} or {resource}
- Redfish® managed resources can have further child resources, which can be accessed by their individual URIs as shown in the URI patterns above. Possible URI paths for a particular Redfish® data model are specified in the Redfish® data model specification [3] for each object. The Redfish® service also offers multiple sub-services, such as, Account Service, Task Service, Composition Service etc. These services and their corresponding resources are also accessible via respective URIs as illustrated in above URI patterns.

NOTE: The Redfish® services, e.g. Task Service, Event Service, etc., are essentially Redfish® resources with their schema defined in the Redfish® data model specification [3].

Redfish® interface [2] enables clients to perform basic CRUD operations on Redfish® resources. Standard HTTP methods like POST, GET, PUT, PATCH and DELETE are used to perform these CRUD operations. Some specific resources also support additional non-CRUD operations, like performing actions on those resources, e.g. restarting a computer system. HTTP POST method is used to perform such actions.

Redfish® interface [2] also enables sending push notifications to subscribed consumers. HTTP POST method is used for this purpose.

4.3.3 Data Model Concepts

The Redfish® managed resources strictly follow the schema specified in the Redfish® data model specification [3]. A Redfish® resource accessed over the Redfish® RESTful interface [2] contains the following properties:

- **@odata.id:** URI of the resource.
- **@odata.type:** Resource type.
- **Id:** Resource identifier.
- **Name:** Resource name.

- Other properties defined in the data model schema of that resource.

A collection of Redfish® resources accessed over the Redfish RESTful interface [2] usually contains the following properties:

- **@odata.id:** URI of the resource.
- **@odata.type:** Resource type.
- **Name:** Resource name.
- **Members:** URIs of the resources that make up the collection.
- **Members@odata.count:** Number of resources that make up the collection.

A mapping of relevant Redfish® resources against the PIM information model is provided in clause 5 of the present document.

5 Mapping of NFV object model to profiled solution objects

5.1 Overview

ETSI GS NFV-IFA 053 [1] specifies a generic information model for a physical resource that works on different granularities. A physical resource can either be individual hardware resources of different types like compute, network, storage, e.g. a memory device, a processor, a switch, disk drive, or an infrastructure resource consisting of various individual hardware components, e.g. a physical server comprising of CPU, memory, NIC and storage disk drives. Similarly, a physical resource can also be a logical system which is composed of different hardware resources, e.g. a full computer system enclosed within a physical server.

The Redfish® data model specification [3] contains schema for various hardware and logical components that can make up a physical resource in the NFVI. The following clauses provide a mapping of relevant objects from Redfish® data model specification [3] for relevant service interfaces exposed by the PIM.

Data objects and their relevant Redfish® counterpart objects, which are common between different PIM services, such as Subscription, Notification etc., are also specified in the subsequent clauses.

5.2 Common data objects

5.2.1 Subscription

Table 5.2.1-1 lists down the properties of the Redfish® object that is mapped to Subscription data object.

Table 5.2.1-1: Redfish® resource object for Subscription

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
EventDestination	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	Represents the target destination for an event subscription and includes type of events that the destination (consumer) is subscribed to.

5.2.2 Notification

Table 5.2.2-1 lists down the properties of the Redfish® object that is mapped to Notification data object.

Table 5.2.2-1: Redfish® resource object for Notification

Redfish® resource object	Redfish® resource object description
Event	Represents the notification payload received by a notification endpoint (target destination), which has subscribed to an event notification.

5.2.3 Task

Table 5.2.3-1 lists down the properties of the Redfish® object that is mapped to the Task object.

Table 5.2.3-1: Redfish® resource object for Task

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
Task	/redfish/v1/TaskService/Tasks/{TaskId}	Represents a task that the Redfish task service schedules or executes.

5.3 Mapping of PIM information model with Redfish® objects

5.3.1 Physical resource object

Table 5.3.1-1 lists down the Redfish® objects that map to the physical resource information model specified as PhysicalResourceInfo in ETSI GS NFV-IFA 053 [1].

Table 5.3.1-1: Redfish® resource objects mapped to NFV's physical resource

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
ComputerSystem	/redfish/v1/Systems/{ComputerSystemId}	Represents a computer system managed by the Redfish® service. Details of its subsystems, such as memory, CPU, storage and network devices are also contained within this resource representation.

5.3.2 Resource zone object

Table 5.3.2-1 lists down the Redfish® objects that map to the resource zone information model specified as ResourceZone in ETSI GS NFV-IFA 053 [1].

Table 5.3.2-1: Redfish® resource objects mapped to NFV's resource zone

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
Zone	/redfish/v1/CompositionService/ResourceZones/{ZoneId}	Represents a resource zone.

5.3.3 Physical resource pool object

Table 5.3.3-1 lists down the Redfish® objects that map to the physical resource pool information model specified as PhysicalResourcePool in ETSI GS NFV-IFA 053 [1].

Table 5.3.3-1: Redfish® resource objects mapped to NFV's physical resource pool

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
ResourceBlock	/redfish/v1/CompositionService/ResourceBlocks/{ResourceBlockId}	Represents a resource block, its components and affinity to composed devices.

5.3.4 NFVI-PoP object

Table 5.3.4-1 lists down the Redfish® objects that map to the NFVI-PoP information model specified as NfviPopInfo in ETSI GS NFV-IFA 053 [1]. Since the NFVI-PoP information model contains a lot of information about various aspects of physical resources in the NFVI, a combination of Redfish objects is needed to provide the corresponding information.

Table 5.3.4-1: Redfish® resource objects mapped to NFV's NFVI-PoP information model

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
Chassis Collection	/redfish/v1/Chassis	Represents a collection of all the available chassis managed by the Redfish® service. Each chassis is represented by the Chassis schema in the Redfish® data model. The Chassis schema represents the sheet-metal confined spaces and logical zones such as racks, enclosures, chassis and all other containers. It also contains information about thermal subsystems, fans, sensors, power supplies, and information about computer systems contained within said Chassis, equipment related to compute, network, storage hardware resources, etc.
Systems Collection	/redfish/v1/Systems	Represents a collection of all the available systems managed by the Redfish® service. Each system is represented by the ComputerSystem schema in the Redfish® data model, which contains information about the computer system itself as well as its various subsystems, such as memory, CPU, storage and network devices that make up said computer system.
Managers Collection	/redfish/v1/Managers	Represents a collection of all the available managers in the Redfish® service. Each manager is represented by the Manager schema in the Redfish® data model, which contains information about the firmware version of the manager, it is current software version, etc. Examples of managers are BMCs (baseboard management controllers), enclosure managers, management controllers, and other subsystems that are assigned manageability functions.
Software inventory Collection	/redfish/v1/UpdateService/FirmwareInventory /redfish/v1/UpdateService/LocalImageStore /redfish/v1/UpdateService/SoftwareInventory (see note)	Represents a collection of the software inventory kept by the Redfish® service. Each software inventory represented by the SoftwareInventory schema in the Redfish® data model. The SoftwareInventory schema contains an inventory of software components. This can include software components such as BIOS, BMC firmware, firmware for other devices, system drivers, or provider software.
Power Equipment Collection	/redfish/v1/PowerEquipment	Represents a collection of all the power equipment's managed by the Redfish® service. Each power equipment is represented by the PowerEquipment schema the Redfish® data model.

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
Cables Collection	/redfish/v1/Cables	Represents a collection of all the cables managed by the Redfish® service. Each cable is represented by the Cable schema in the Redfish® data model, which contains properties that describe a cable connecting endpoints of a chassis, port, or any other cable-compatible endpoint.
NOTE: The software inventory collections available at all the listed Redfish® resource URIs may be considered for the overall software inventory collection associated with an NFVI-PoP.		

5.4 Performance management related objects

5.4.1 PMJob

Table 5.4.1-1 lists down the properties of the Redfish® object that is mapped to the PmJob object.

Table 5.4.1-1: Redfish® resource object for PmJob

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
MetricReportDefinition	/redfish/v1/TelemetryService/MetricReportDefinitions/{MetricReportDefinitionId}	Represents a set of metrics that are to be collected in a metric report.

5.4.2 Threshold

Table 5.4.2-1 lists down the properties of the Redfish® object that is mapped to the Threshold object.

Table 5.4.2-1: Redfish® resource object for Threshold

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
Trigger	/redfish/v1/TelemetryService/Triggers/{TriggerId}	Represents trigger conditions and thresholds that apply to the metrics being collected.

5.4.3 PerformanceReport

Table 5.4.3-1 lists down the properties of the Redfish® object that is mapped to the PerformanceReport object.

Table 5.4.3-1: Redfish® resource object for PerformanceReport

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
MetricReport	/redfish/v1/TelemetryService/MetricReports/{MetricReportId}	Represents a set of collected metrics and their values as part of a performance report.

5.5 Log management related objects

5.5.1 LoggingJob

Table 5.5.1-1 lists down the properties of the Redfish® object that is mapped to the LoggingJob object.

Table 5.5.1-1: Redfish® resource object for LoggingJob

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
LogService	.../LogServices/{LogServiceId}	Represents a log service for a Redfish implementation and contains properties for monitoring and configuring a logging job.

Possible URI paths for the LogService object are specified in the Redfish® data model specification [3].

5.5.2 Log

Table 5.5.2-1 lists down the properties of the Redfish® object that is mapped to the Log object.

Table 5.5.2-1: Redfish® resource object for LoggingJob

Redfish® resource object	Redfish® resource URI	Redfish® resource object description
LogEntry	.../LogServices/{LogServiceId}/Entries/{LogEntryId}	Represents the format of log collected by a log service in the Redfish implementation.

Possible URI paths for the LogEntry object are specified in the Redfish® data model specification [3].

5.6 Fault management related objects

5.6.1 Alarm

Table 5.6.1-1 lists down the properties of the Redfish® object that is mapped to Alarm data object.

Table 5.6.1-1: Redfish® resource object for Alarm

Redfish® resource object	Redfish® resource object description
Event	Represents a notification containing the alarm, which is received by the consumer that has subscribed to receive alarms associated with physical resources.

6 Physical resource provisioning and lifecycle management service interface

6.1 Description

The physical resource provisioning and lifecycle management service interface allows the API consumers to invoke provisioning and lifecycle management operations on managed physical resources. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Create a physical resource object
- Retrieve information about a physical resource object
- Retrieve information about a collection of physical resource objects
- Update information about a physical resource object
- Perform an action on a physical resource object
- Delete a physical resource object

- Subscribe to notifications related to events pertaining physical resource objects
- Notify subscribers according to the subscriptions

6.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1/Systems
- /redfish/v1/EventService/Subscriptions
- /redfish/v1/TaskService/TaskMonitors

6.3 Resource structure and HTTP methods

Figures 6.3-1, 6.3-2 and 6.3-3 show resource structures of the Redfish® API [2] resources that are applicable for the physical resource provisioning and lifecycle management service interface.

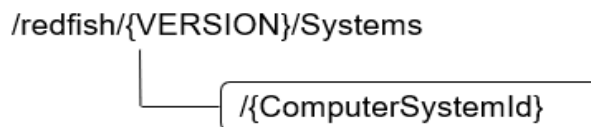


Figure 6.3-1: API resource structure of Physical Resource object for the physical resource provisioning and lifecycle management service interface



Figure 6.3-2: API resource structure of Subscription object for physical resource provisioning and lifecycle management service interface

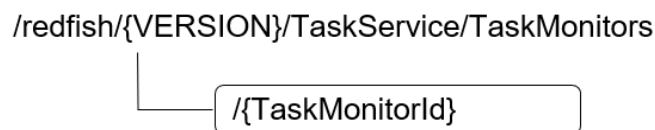


Figure 6.3-3: API resource structure of the Individual Task Monitor resource for physical resource provisioning and lifecycle management service interface

Table 6.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 6.3-1 shall be supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition. The HTTP methods marked as "O" (Optional) in the "Cat" column may be optionally supported by the PIM.

Table 6.3-1: API Resources and HTTP methods of physical resource provisioning and lifecycle management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Systems	/redfish/v1/Systems	GET	M	Get a collection of Computer Systems.
		POST	M	Create a new "Individual Computer System" resource.
Individual Computer System	/redfish/v1/Systems/{ComputerSystemId}	GET	M	Get information about the "Individual Computer System" resource.
		PATCH	M	Modify information of an "Individual Computer System" resource.
		PUT	O	Replace an "Individual Computer System" resource.
		DELETE	M	Delete an "Individual Computer System" resource.
Individual Computer System Action	/redfish/v1/Systems/{ComputerSystemId}/Actions/ComputerSystem.{ActionName}	POST	M	Performs a qualified action on the "Individual Computer System" resource.
Subscriptions	/redfish/v1/EventService/Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 1).
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 2)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer.				
NOTE 2: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

6.4 Sequence diagrams (informative)

6.4.1 Introduction

Flows related to task monitoring, subscription management and notifications are described in Annex A of the present document.

6.4.2 Creation of a Computer System resource

This clause describes a sequence for creating an individual Computer System resource.

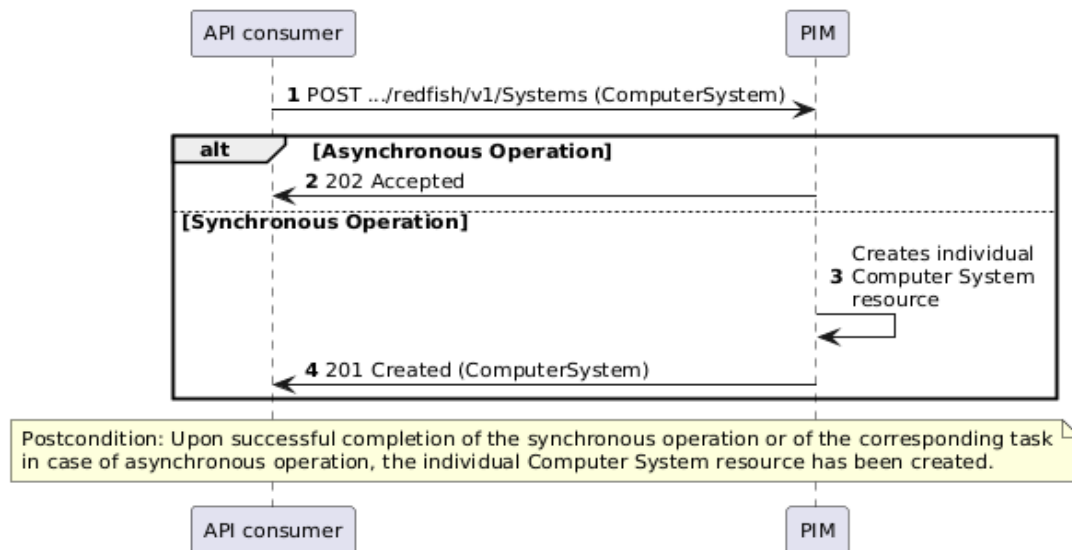


Figure 6.4.2-1: Flow of Computer System creation

The creation of a Computer System resource, as illustrated in Figure 6.4.2-1, consists of the following steps.

Precondition: None.

1. The API consumer sends a POST request to the Systems resource including a valid representation of the individual Computer System resource in the message content body.
2. If the PIM supports asynchronous operations and creation of the computer system resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM creates an individual Computer System resource.
4. The PIM returns a "201 Created" response to the API consumer and includes in the message content body a representation of the created Computer System resource. The response also contains the URI of the newly created resource in the Location header.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual Computer System resource has been created.

Error handling: In case of failure, appropriate error information is provided in the response.

6.4.3 Querying information about Computer System resources

This clause describes the sequences for querying information about both Computer System resource collection and individual Computer System resource.

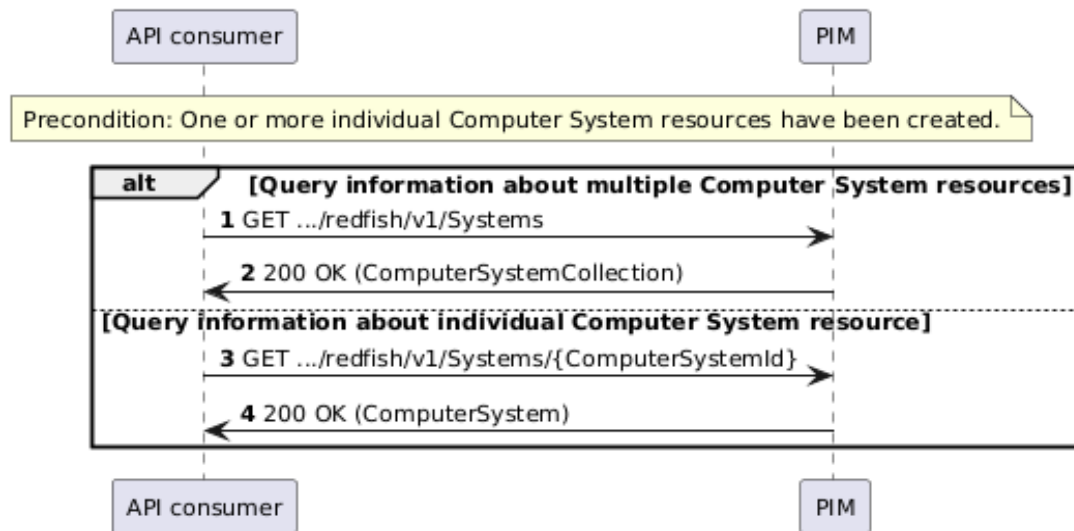


Figure 6.4.3-1: Flow of querying information about Computer System resources

The querying of information about one or more Computer System resources, as illustrated in Figure 6.4.3-1, consists of the following steps.

Precondition: One or more individual Computer System resources have been created.

1. If the API consumer intends to get information about multiple Computer System resources, it sends a GET request to the Systems resource.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a Computer System collection, containing information about all the individual Computer System resources that have been created.
3. If the API consumer intends to get information about an individual Computer System resource, it sends a GET request to the Individual Computer System resource identified by its name in the URI.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual Computer System resource.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

6.4.4 Modifying a Computer System resource

This clause describes a sequence for modifying an individual Computer System resource.

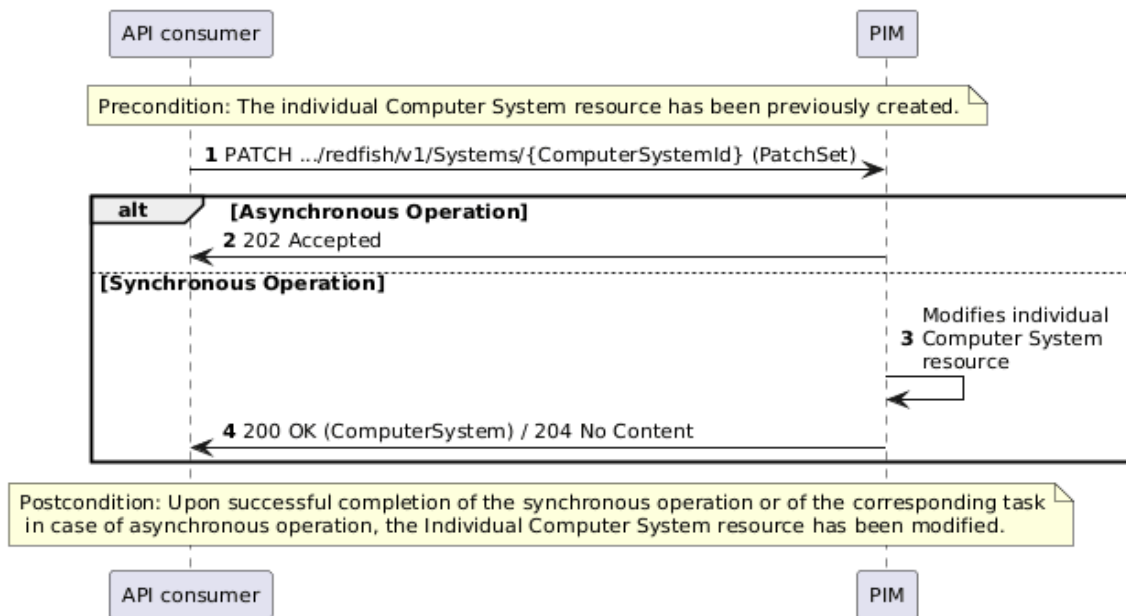


Figure 6.4.4-1: Flow of Computer System modification

The modification of an individual Computer System resource, as illustrated in Figure 6.4.4-1, consists of the following steps.

Precondition: The individual Computer System resource has been previously created.

1. The API consumer sends a PATCH request to the Individual Computer System resource, including data structure in the message content body representing the modifiable properties to be modified in the individual computer system resource.
2. If the PIM supports asynchronous operations and modification of the computer system resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM modifies the Individual Computer System resource.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified Computer System resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Computer System resource has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

6.4.5 Replacing a Computer System resource

This clause describes a sequence for replacing an individual Computer System resource.

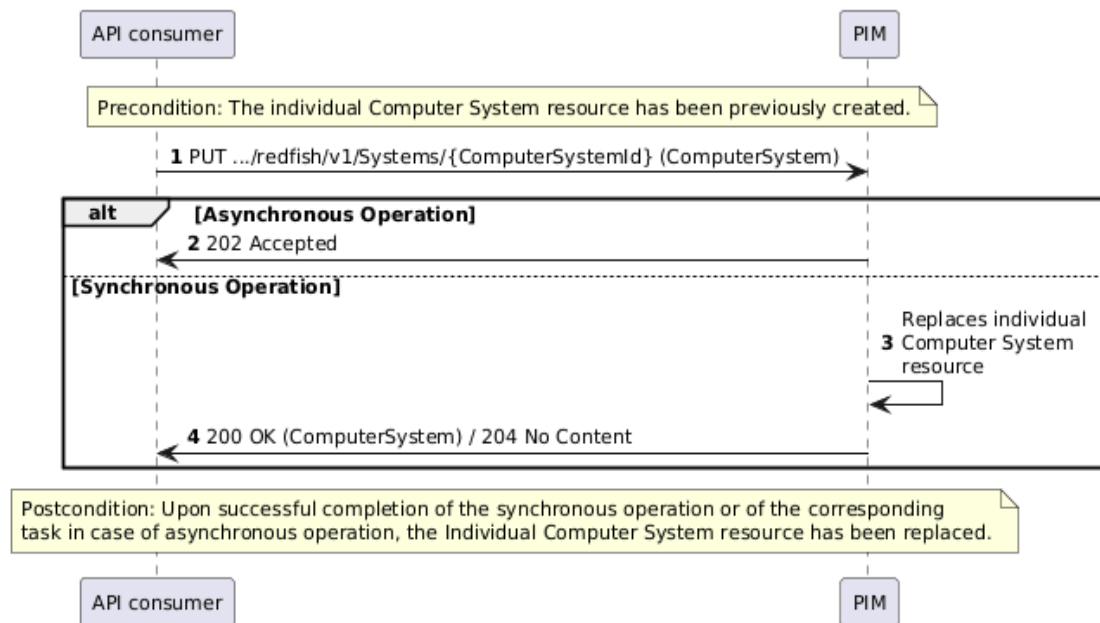


Figure 6.4.5-1: Flow of Computer System replacement

The replacement of an individual Computer System resource, as illustrated in Figure 6.4.5-1, consists of the following steps.

Precondition: The individual Computer System resource has been previously created.

1. The API consumer sends a PUT request to the Individual Computer System resource including a valid representation of the individual Computer System resource to be replaced in the message content body.
2. If the PIM supports asynchronous operations and replacement of the computer system resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM replaces the Individual Computer System resource.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced Computer System resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Computer System resource has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

6.4.6 Deleting a Computer System resource

This clause describes a sequence for deleting an Individual Computer System resource.

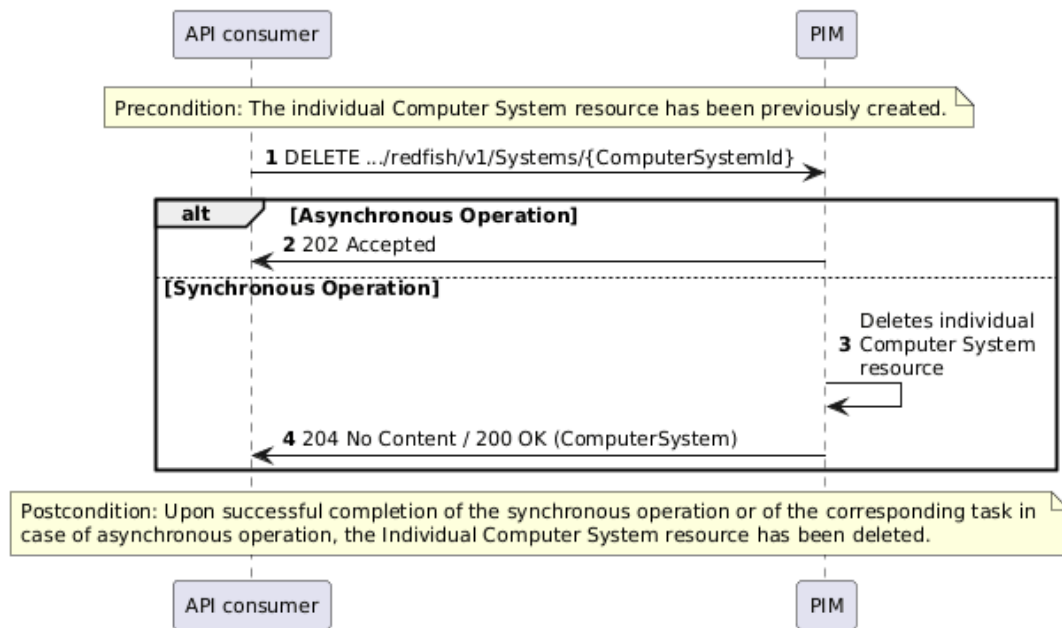


Figure 6.4.6-1: Flow of Computer System deletion

The deletion of an individual Computer System resource, as illustrated in Figure 6.4.6-1, consists of the following steps.

Precondition: The individual Computer System resource has been previously created.

1. The API consumer sends a DELETE request to the Individual Computer System resource.
2. If the PIM supports asynchronous operations and deletion of the computer system resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM deletes the individual Computer System resource.
4. The PIM returns a "204 No Content" response to the API consumer indicating that the resource has been successfully deleted. Alternatively, the PIM can also return a "200 OK" response to the API consumer including a representation of the deleted Computer System resource in the message content body.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Computer System resource has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

6.4.7 Perform an action on a Computer System resource

This clause describes a sequence for performing an action on an Individual Computer System resource.

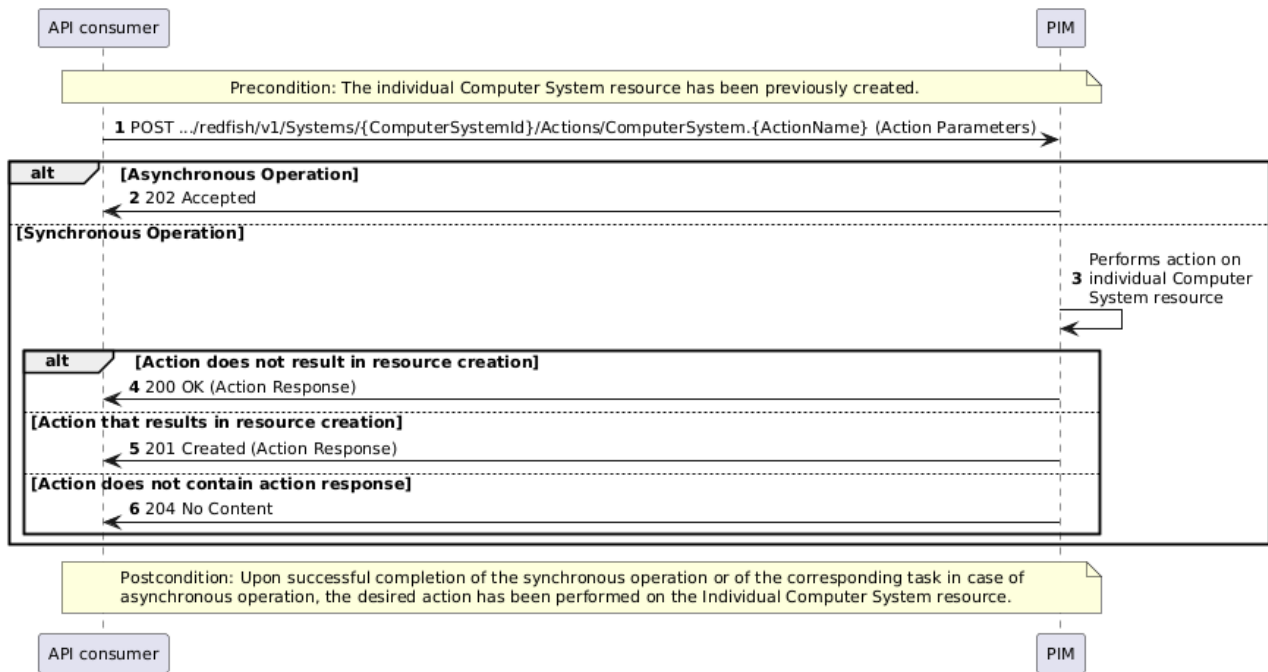


Figure 6.4.7-1: Flow of performing an action on a Computer System resource

Performing of an action on an individual Computer System resource, as illustrated in Figure 6.4.7-1, consists of the following steps.

Precondition: The individual Computer System resource has been previously created.

1. The API consumer sends a POST request to the Individual Computer System Action resource, indicating a valid action name that is supported by the individual Computer System in the URI. If applicable, the request body also contains parameters for the action.
2. If the PIM supports asynchronous operations and the desired action requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM performs the desired action on the individual Computer System resource.
4. If the desired action does not result in creation of a new resource, the PIM returns a "200 OK" response to the API consumer indicating that the action been performed successfully. The corresponding action response is provided in the response body.
5. If the desired action results in creation of some new resource, the PIM returns a "201 Created" response to the API consumer indicating that the action been performed successfully. If applicable, action response is provided in the response body. The response also contains the URI of the newly created resource in the Location header.
6. If the desired action does not result in creation of a new resource and the action's schema does not contain any action response, the PIM returns a "204 No Content" response to the API consumer indicating that the action been performed successfully.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the desired action has been performed on the Individual Computer System resource.

Error handling: In case of failure, appropriate error information is provided in the response.

6.5 Resources

6.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource provisioning and lifecycle management service interface.

6.5.2 Resource: Systems

This resource represents computer systems as composed physical resources managed by the PIM and can be used to retrieve a collection of available computer system resources as well as to create an individual computer system resource.

Table 6.5.2-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.2-1: Systems resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Systems (see note)	GET	Lists a collection of Computer System resources. Retrieves information about the composed physical resources (computer systems) managed by the PIM.	PimPrLcmMgt.002 PimPrLcmMgt.003
	POST	Creates a new "Individual Computer System" resource, representing a composed physical resource.	PimPrLcmMgt.001 PimPrLcmMgt.003
NOTE: Refer to Table 6.3-1 for the full URI of this resource.			

6.5.3 Resource: Individual Computer System

This resource represents an individual computer system as a composed physical resource managed by the PIM and can be used to query, update or delete an individual computer system resource.

Table 6.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.3-1: Individual Computer System resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Systems/{Computer SystemId} (see note)	GET	Retrieves information about a composed physical resource, i.e. individual computer system, managed by the PIM.	PimPrLcmMgt.002 PimPrLcmMgt.003
	PUT	Modifies the individual computer system resource by replacing it whole.	PimPrLcmMgt.003
	PATCH	Modifies the individual computer system resource by updating only a patch of it.	PimPrLcmMgt.003 PimPrLcmMgt.004
	DELETE	Deletes the individual computer system resource.	PimPrLcmMgt.003
NOTE: Refer to Table 6.3-1 for the full URI of this resource.			

6.5.4 Resource: Individual Computer System Action

This resource can be used to perform an action on an individual computer system resource.

Table 6.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.4-1: Individual Computer System Action resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Systems/{ComputerSystemId}/Actions/ComputerSystem.{ActionName} (see note 1)	POST	Performs an action on "Individual Computer System" resource, representing a composed physical resource (see note 2).	PimPrLcmMgt.001 PimPrLcmMgt.003 PimPrLcmMgt.004
NOTE 1: Refer to Table 6.3-1 for the full URI of this resource.			
NOTE 2: The parameter {ActionName} depends on the desired action to be performed on the Computer System resource. Its value can be one of the actions supported by the ComputerSystem schema as specified in the Redfish data model specification [3], e.g. Reset, Decommission, AddResourceBlock, SetDefaultBootOrder, etc.			

6.5.5 Resource: Subscriptions

This resource represents the subscriptions related to physical resources managed by the PIM and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource.

Table 6.5.5-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.5-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of Subscription resources. Retrieves information about the created subscriptions related to provisioning and lifecycle management of physical resources managed by the PIM.	PimPrLcmMgt.005
	POST	Creates a new "Individual Subscription" resource.	PimPrLcmMgt.005
NOTE: Refer to Table 6.3-1 for the full URI of this resource.			

6.5.6 Resource: Individual Subscription

This resource represents an individual subscription related to provisioning and lifecycle management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resources.

Table 6.5.6-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.6-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription managed by the PIM.	PimPrLcmMgt.005
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrLcmMgt.005
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrLcmMgt.005
	DELETE	Deletes the individual subscription resource.	PimPrLcmMgt.005
NOTE: Refer to Table 6.3-1 for the full URI of this resource.			

6.5.7 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 6.5.7-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 6.5.7-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource provisioning and lifecycle management once that operation has been completed.	PimPrLcmMgt.006
NOTE: Refer to Table 6.3-1 for the full URI of this resource.			

6.6 Data model

The request and response data structures of the physical resource provisioning and lifecycle management service interface are defined in the Redfish® data model specification [3].

7 Physical resource inventory management service interface

7.1 Description

The physical resource inventory management service interface allows the API consumers to perform inventory management operations related to physical resources in the NFVI. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Retrieve information about physical resource inventory of an NFVI-PoP.
- Update inventory information of an NFVI-PoP.
- Subscribe to notifications related to changes in inventory.
- Notify subscribers according to the subscriptions.

7.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1
- /redfish/v1/EventService/Subscriptions
- /redfish/v1/TaskService/TaskMonitors

7.3 Resource structure and HTTP methods

The inventory of physical resources within an NFVI-PoP managed by the PIM are represented by different collections of Redfish® objects. These Redfish® collections, along with their resource URIs, that provide the corresponding information about physical resources in an NFVI-PoP are specified in clause 5.3.4 of the present document.

For the sake of simplicity and to avoid repetition, the present clause denotes each Redfish® collection as "InventoryObjectCollection" and each individual object of a collection as "IndividualInventoryObject".

Figures 7.3-1, 7.3-2 and 7.3-3 show resource structures of the Redfish® API [2] resources that are applicable for the physical resource inventory management service interface.

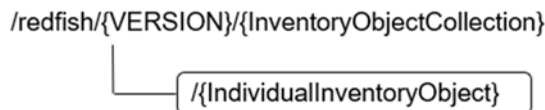


Figure 7.3-1: API resource structure of "InventoryObjectCollection" resource for the physical resource inventory management service interface



Figure 7.3-2: API resource structure of Subscription resource for physical resource inventory management service interface

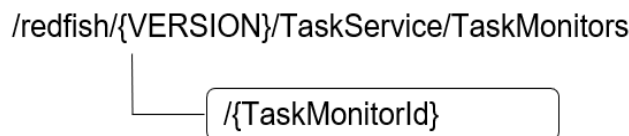


Figure 7.3-3: API resource structure of the Individual Task Monitor resource for physical resource inventory management service interface

Table 7.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 7.3-1 shall be supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition. The HTTP methods marked as "O" (optional) in the "Cat" column may be optionally supported by the PIM.

Table 7.3-1: API Resources and HTTP methods of physical resource inventory management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
InventoryObjectCollection	/redfish/v1/{InventoryObjectCollection} (see note 1)	GET	M	Get a collection of inventory objects of same kind.
IndividualInventoryObject	/redfish/v1/{InventoryObjectCollection}/{IndividualInventoryObject} (see note 2)	GET	M	Get an individual object from a collection of inventory objects of same kind.
		PATCH	M	Modify an individual object from an inventory collection.
		PUT	O	Replace an individual object from an inventory collection.
Subscriptions	/redfish/v1/EventService/Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 3). This resource is used to send information about faults to the consumer in the form of an alert type event.
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 4)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: See clause 5.3.4 for the resource URIs of each inventory object collection. NOTE 2: The resource URI of an individual object can be acquired from the corresponding object collection in its "Members" property. NOTE 3: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer. NOTE 4: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

7.4 Sequence diagrams (informative)

7.4.1 Introduction

The following clauses describe the sequences for querying, replacing and modifying an inventory object.

The diagrams and their description contain placeholders indicated as <inventory object collection URI> and <inventory object URI> which refer to the URIs of the corresponding API resources specified in clause 7.3.

Flows related to task monitoring, subscription management and notifications are described in Annex A of the present document.

7.4.2 Querying information about inventory objects

This clause describes the sequences for querying information about both multiple and individual inventory objects.

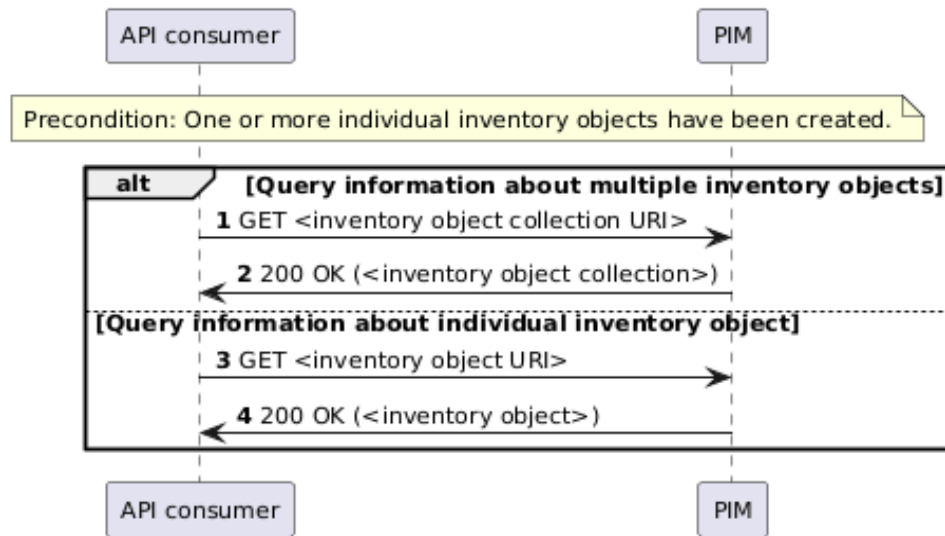


Figure 7.4.2-1: Flow of querying information about inventory objects

The querying of information about one or more inventory objects, as illustrated in Figure 7.4.2-1, consists of the following steps.

Precondition: One or more individual inventory objects have previously been created.

1. If the API consumer intends to get information about multiple inventory objects, it sends a GET request to the appropriate <inventory object collection URI>.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of an <inventory object collection>, containing information about all the individual inventory objects that have been created.
3. If the API consumer intends to get information about an individual inventory object, it sends a GET request to the appropriate <inventory object URI>.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual inventory object.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

7.4.3 Modifying an inventory object

This clause describes a sequence for modifying an individual inventory object.

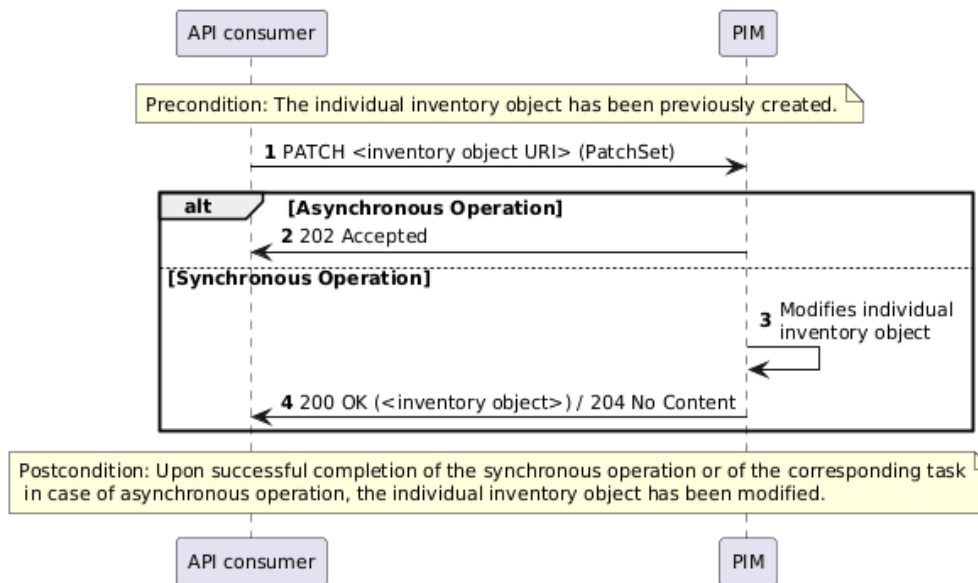


Figure 7.4.3-1: Flow of inventory object modification

The modification of an inventory object, as illustrated in Figure 7.4.3-1, consists of the following steps.

Precondition: The individual inventory object has been previously created.

1. The API consumer sends a PATCH request to the <inventory object URI>, including data structure in the message content body representing the modifiable properties to be modified in the individual inventory object.
2. If the PIM supports asynchronous operations and modification of the inventory object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM modifies the individual inventory object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified inventory object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual inventory object has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

7.4.4 Replacing an inventory object

This clause describes a sequence for replacing an individual inventory object.

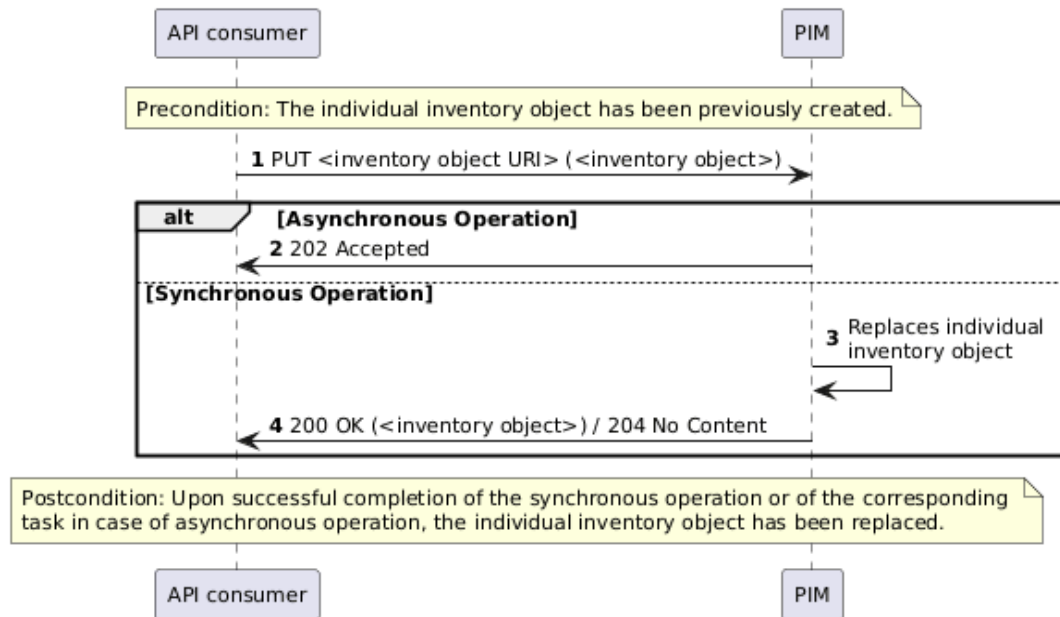


Figure 7.4.4-1: Flow of inventory object replacement

The replacement of an individual inventory object, as illustrated in Figure 7.4.4-1, consists of the following steps.

Precondition: The individual inventory object has been previously created.

1. The API consumer sends a PUT request to the <inventory object URI> including a valid representation of the individual inventory object to be replaced in the message content body.
2. If the PIM supports asynchronous operations and replacement of the inventory object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM replaces the individual inventory object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced inventory object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual inventory object has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

7.5 Resources

7.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource inventory management service interface.

7.5.2 Resource: InventoryObjectCollection

This resource represents a collection of physical resource inventory objects of the same kind, e.g. a chassis collection, systems collection, cables collection, software inventory collection, etc. The API resource can be used to retrieve a collection of physical resource inventory objects that are managed by the PIM.

Table 7.5.2-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 7.5.2-1: InventoryObjectCollection resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../{InventoryObjectCollection} (see note)	GET	Get a collection of inventory objects Retrieves information about the inventory of physical resources within an NFVI managed by the PIM	PimPrImMgt.001
NOTE: Refer to Table 7.3-1 for the full URI of this resource.			

7.5.3 Resource: IndividualInventoryObject

This resource represents an individual physical resource inventory object managed by the PIM, and can be used to query or update the individual object.

Table 7.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 7.5.3-1: IndividualInventoryObject resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../{InventoryObjectCollection}/{IndividualInventoryObject} (see note)	GET	Retrieves information about the individual physical resource inventory object	PimPrImMgt.001
	PUT	Modifies the individual inventory object by replacing it whole.	PimPrImMgt.004
	PATCH	Modifies the individual inventory object by updating only a patch of it.	PimPrImMgt.004
NOTE: Refer to Table 7.3-1 for the full URI of this resource.			

7.5.4 Resource: Subscriptions

This resource represents subscriptions related to inventory management of physical resources and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource. This resource can be used to create a subscription for receiving notifications about any changes in the inventory information of physical resources.

Table 7.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 7.5.4-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of subscriptions. Retrieves information about the created subscriptions related to inventory management of physical resources managed by the PIM.	PimPrImMgt.003
	POST	Creates a new Individual Subscription resource.	PimPrImMgt.003
NOTE: Refer to Table 7.3-1 for the full URI of this resource.			

7.5.5 Resource: Individual Subscription

This resource represents an individual subscription related to inventory management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resource.

Table 7.5.5-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 7.5.5-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription resource managed by the PIM.	PimPrImMgt.003
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrImMgt.003
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrImMgt.003
	DELETE	Deletes the individual subscription resource.	PimPrImMgt.003
NOTE: Refer to Table 7.3-1 for the full URI of this resource.			

7.5.6 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 7.5.6-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 7.5.6-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource log management once that operation has been completed.	PimPrImMgt.003 PimPrImMgt.004
NOTE: Refer to Table 7.3-1 for the full URI of this resource.			

7.6 Data model

The request and response data structures of the physical resource inventory management service interface are defined in the Redfish® data model specification [3].

8 Physical resource topology management service interface

8.1 Description

The physical resource topology management service interface allows the API consumers to invoke topology management operations related to physical infrastructure. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Create a topology object.
- Retrieve information about a topology object.
- Retrieve information about a collection of topology objects.
- Update information about a topology object.
- Delete a topology object.
- Subscribe to notifications related to events pertaining topology objects.
- Notify subscribers according to the subscriptions.

NOTE: The topology object in the above list refers to both the resource zone and physical resource pool objects specified in clause 5 of the present document.

8.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1/CompositionService/ResourceZones
- /redfish/v1/CompositionService/ResourceBlock
- /redfish/v1/EventService/Subscriptions
- /redfish/v1/TaskService/TaskMonitors

8.3 Resource structure and HTTP methods

Figures 8.3-1, 8.3-2, 8.3-3 and 8.3-4 show resource structures of the Redfish® API [2] resources that are applicable for the physical resource topology management service interface.

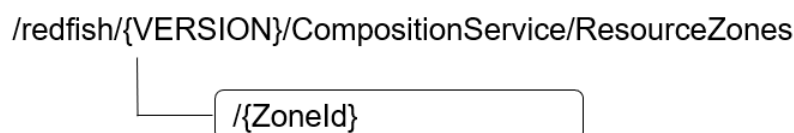


Figure 8.3-1: API resource structure of Resource Zone resource for the physical resource topology management service interface

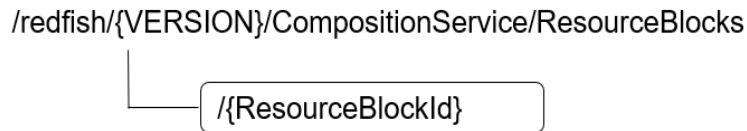


Figure 8.3-2: API resource structure of Physical Resource Pool resource for the physical resource topology management service interface



Figure 8.3-3: API resource structure of Subscription resource for physical resource topology management service interface

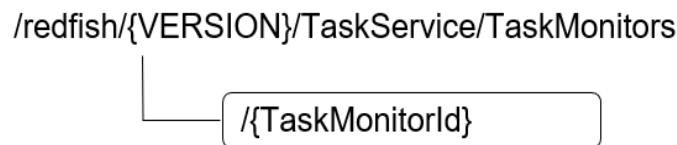


Figure 8.3-4: API resource structure of the Individual Task Monitor resource for physical resource topology management service interface

Table 8.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (Mandatory) in the "Cat" column of Table 8.3-1 shall be supported by the PIM. The HTTP methods marked as "CM" (Conditional Mandatory) shall be supported by the PIM depending on the corresponding condition. The HTTP methods marked as "O" (Optional) in the "Cat" column may be optionally supported by the PIM.

Table 8.3-1: API Resources and HTTP methods of physical resource topology management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
ResourceZones	/redfish/v1/CompositionService/ResourceZones	GET	M	Get a collection of Resource Zones.
		POST	M	Create a new "Individual Resource Zone" resource.
Individual Resource Zone	/redfish/v1/CompositionService/ResourceZones/{ZoneId}	GET	M	Get information about the "Individual Resource Zone" resource.
		PATCH	M	Modify information of an "Individual Resource Zone" resource.
		PUT	O	Replace an "Individual Resource Zone" resource.
		DELETE	M	Delete an "Individual Resource Zone" resource.
Resource Blocks	/redfish/v1/CompositionService/ResourceBlocks	GET	M	Get a collection of Resource Zones.
		POST	M	Create a new "Individual Resource Block" resource.
Individual Resource Block	/redfish/v1/CompositionService/ResourceBlocks/{ResourceBlockId}	GET	M	Get information about the "Individual Resource Block" resource.
		PATCH	M	Modify information of an "Individual Resource Block" resource.
		PUT	O	Replace an "Individual Resource Block" resource.
		DELETE	M	Delete an "Individual Resource Block" resource.
Subscriptions	/redfish/v1/EventService/Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 1).
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 2)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer.				
NOTE 2: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

8.4 Sequence diagrams (informative)

8.4.1 Introduction

The sequence diagrams provided in the subsequent sub-clauses are generalized to indicate management of a general topology object, which refers to both the resource zone and physical resource pool objects specified in clause 5 of the present document.

The diagrams and their description contain placeholders such as <topology object collection URI>, <topology object URI> and <topology object>, which correspond to resource URIs and topology objects specified in clauses 8.3 and 5.3 respectively.

Flows related to task monitoring, subscription management and notifications are described in Annex A of the present document.

8.4.2 Creation of a topology object

This clause describes a sequence for creating an individual resource corresponding to a topology management object.

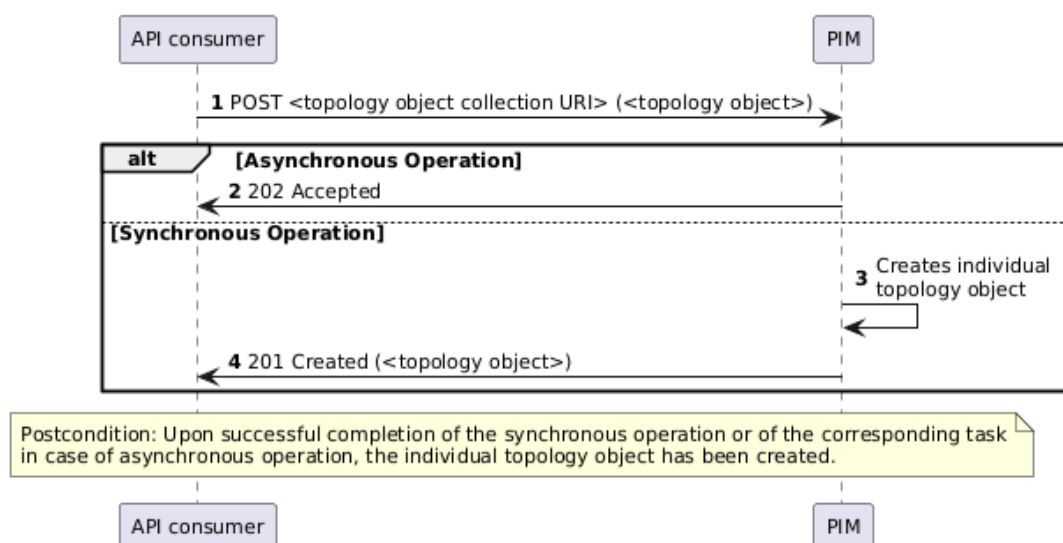


Figure 8.4.2-1: Flow of topology object creation

The creation of a topology object, as illustrated in Figure 8.4.2-1, consists of the following steps.

Precondition: None.

1. The API consumer sends a POST request to the <topology object collection URI> including a valid <topology object> representation in the message content body.
2. If the PIM supports asynchronous operations and creation of the topology object resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM creates an individual topology object.
4. The PIM returns a "201 Created" response to the API consumer and includes in the message content body a representation of the created topology object. The response also contains the URI of the newly created topology object in the Location header.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual topology object has been created.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.3 Querying information about topology objects

This clause describes the sequences for querying information about both multiple and individual topology management objects.

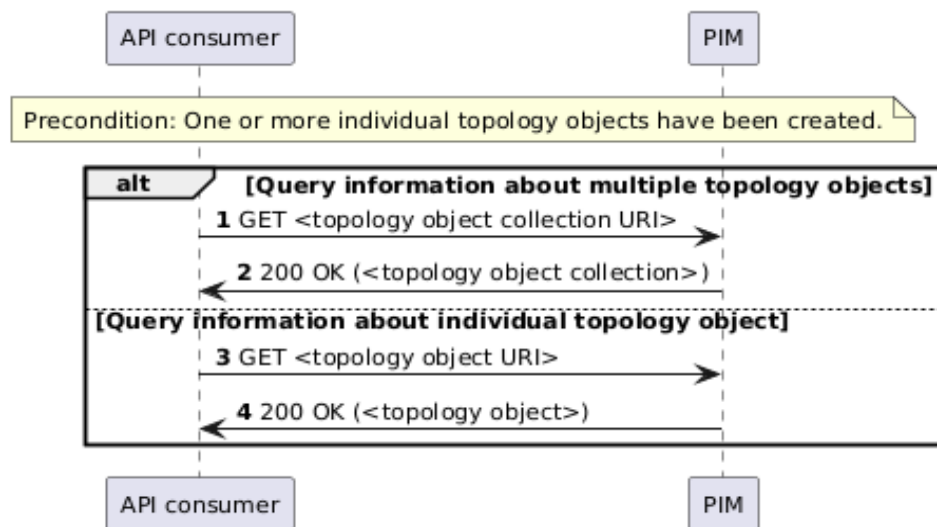


Figure 8.4.3-1: Flow of querying information about topology objects

The querying of information about one or more topology objects, as illustrated in Figure 8.4.3-1, consists of the following steps.

Precondition: One or more individual topology objects have previously been created.

1. If the API consumer intends to get information about multiple topology objects, it sends a GET request to the appropriate <topology object collection URI>.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a <topology object collection>, containing information about all the individual topology objects that have been created.
3. If the API consumer intends to get information about an individual topology object, it sends a GET request to the appropriate <topology object URI>.

4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual topology object.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.4 Modifying a topology object

This clause describes a sequence for modifying an individual topology object.

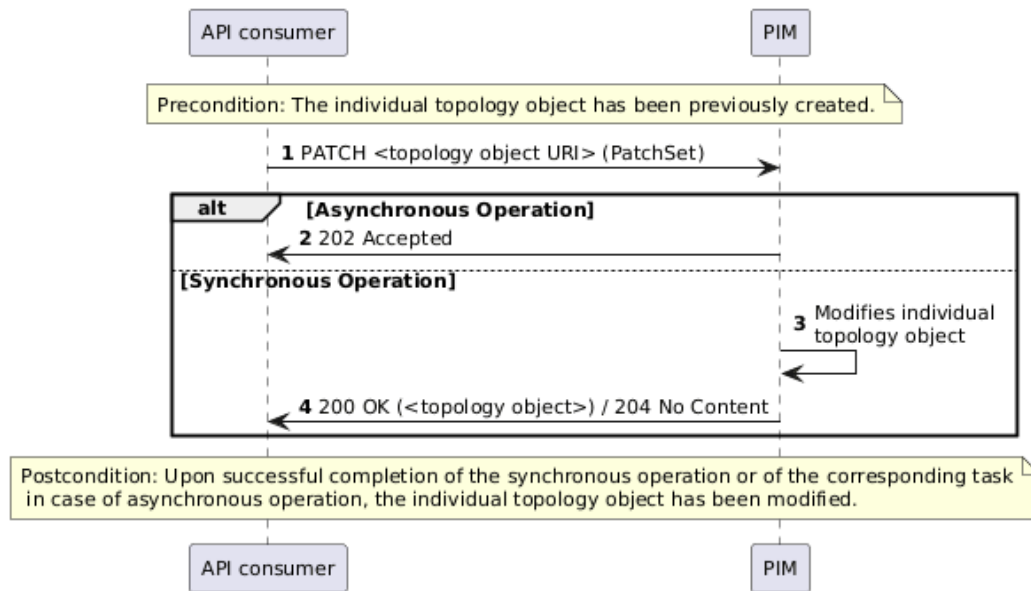


Figure 8.4.4-1: Flow of topology object modification

The modification of a topology object, as illustrated in Figure 8.4.4-1, consists of the following steps. The modification of a topology object, as illustrated in Figure 8.4.4-1, consists of the following steps.

Precondition: The individual topology object has been previously created.

1. The API consumer sends a PATCH request to the <topology object URI>, including data structure in the message content body representing the modifiable properties to be modified in the individual topology object.
2. If the PIM supports asynchronous operations and modification of the topology object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM modifies the individual topology object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified topology object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual topology object has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.5 Replacing a topology object

This clause describes a sequence for replacing an individual topology object.

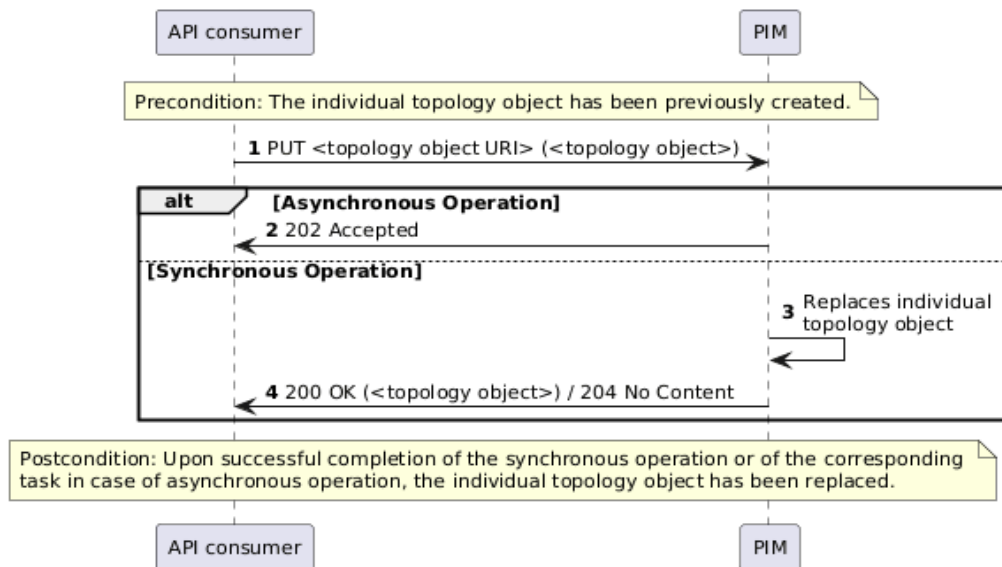


Figure 8.4.5-1: Flow of topology object replacement

The replacement of an individual topology object, as illustrated in Figure 8.4.5-1, consists of the following steps.

Precondition: The individual topology object has been previously created.

1. The API consumer sends a PUT request to the <topology object URI> including a valid representation of the individual topology object to be replaced in the message content body.
2. If the PIM supports asynchronous operations and replacement of the topology object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM replaces the individual topology object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced topology object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual topology object has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.6 Deleting a topology object

This clause describes a sequence for deleting an individual topology object.

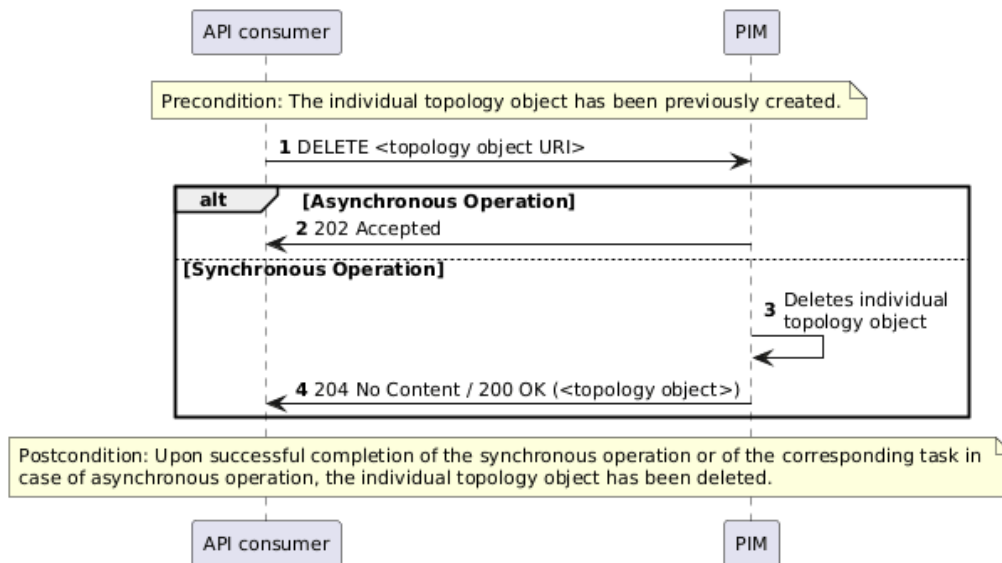


Figure 8.4.6-1: Flow of topology object deletion

The deletion of an individual topology object, as illustrated in Figure 8.4.6-1, consists of the following steps.

Precondition: The individual topology object has been previously created.

1. The API consumer sends a DELETE request to the <topology object URI>.
2. If the PIM supports asynchronous operations and deletion of the topology object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM deletes the individual topology object.
4. The PIM returns a "204 No Content" response to the API consumer indicating that the topology object has been successfully deleted. Alternatively, the PIM can also return a "200 OK" response to the API consumer including a representation of the deleted topology object in the message content body.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual topology object has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

8.5 Resources

8.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource topology management service interface.

8.5.2 Resource: Resource Zones

This resource represents the resource zones within the NFVI managed by the PIM and can be used to retrieve a collection of available resource zones as well as to create an individual resource zone resource.

Table 8.5.2-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 8.5.2-1: Resource Zones resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../ResourceZones (see note)	GET	Lists a collection of available resource zones. Retrieves physical topology information of the physical resources managed by the PIM.	PimPrTmMgt.001 PimPrTmMgt.004
	POST	Creates a new "Individual Resource Zone" resource.	PimPrTmMgt.004
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.3 Resource: Individual Resource Zone

This resource represents an individual resource zone within the NFVI containing physical resources managed by the PIM and can be used to query, update or delete an individual resource zone resource.

Table 8.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 8.5.3-1: Individual Resource Zone resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../ResourceZones/{Zoneld} (see note)	GET	Retrieves information about an individual resource zone managed by the PIM.	PimPrTmMgt.001 PimPrTmMgt.004
	PUT	Modifies the individual resource zone resource by replacing it whole.	PimPrTmMgt.004
	PATCH	Modifies the individual resource zone resource by updating only a patch of it.	PimPrTmMgt.004
	DELETE	Deletes the individual resource zone resource.	PimPrTmMgt.004
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.4 Resource: Resource Blocks

This resource represents the resource blocks corresponding to logical groupings of physical resources managed by the PIM and can be used to retrieve a collection of available resource blocks as well as to create an individual resource block resource.

Table 8.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 8.5.4-1: Resource Blocks resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../ResourceBlocks (see note)	GET	Lists a collection of resource blocks. Retrieves logical topology information of the physical resources managed by the PIM.	PimPrTmMgt.001 PimPrTmMgt.004
	POST	Creates a new "Individual Resource Block" resource.	PimPrTmMgt.004
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.5 Resource: Individual Resource Block

This resource represents an individual resource block representing a logical grouping of physical resources within the NFVI and can be used to query, update or delete an individual resource block resource.

Table 8.5.5-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 8.5.5-1: Individual Resource Block resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../ResourceBlocks/{ResourceId} (see note)	GET	Retrieves information about an individual resource block managed by the PIM.	PimPrTmMgt.001 PimPrTmMgt.004
	PUT	Modifies the individual resource block resource by replacing it whole.	PimPrTmMgt.004
	PATCH	Modifies the individual resource block resource by updating only a patch of it.	PimPrTmMgt.004
	DELETE	Deletes the individual resource block resource.	PimPrTmMgt.004
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.6 Resource: Subscriptions

This resource represents subscriptions related to topology management of physical resources and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource.

Table 8.5.6-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 8.5.6-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of subscriptions. Retrieves information about the created subscriptions related to topology management of physical resources managed by the PIM.	PimPrTmMgt.003
	POST	Creates a new "Individual Subscription" resource.	PimPrTmMgt.003
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.7 Resource: Individual Subscription

This resource represents an individual subscription related to topology management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resources.

Table 8.5.7-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 8.5.7-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription resource managed by the PIM.	PimPrTmMgt.003
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrTmMgt.003
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrTmMgt.003
	DELETE	Deletes the individual subscription resource.	PimPrTmMgt.003
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.5.8 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 8.5.8-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 8.5.8-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource topology management once that operation has been completed.	PimPrTmMgt.004
NOTE: Refer to Table 8.3-1 for the full URI of this resource.			

8.6 Data model

The request and response data structures of the physical resource topology management service interface are defined in the Redfish® data model specification [3].

9 Physical resource performance management service interface

9.1 Description

The physical resource performance management service interface allows the API consumers to invoke Performance Management (PM) operations related to physical resources. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Create, update and delete a PM Job.
- Retrieve PM Jobs.
- Create, update and delete a Threshold.
- Retrieve Thresholds.
- Retrieve Performance Reports.
- Subscribe to notifications related to performance reporting events.
- Notify subscribers according to the subscriptions.

9.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1/TelemetryService/MetricReportDefinitions
- /redfish/v1/TelemetryService/Triggers
- /redfish/v1/TelemetryService/MetricReports
- /redfish/v1/EventService/Subscriptions

9.3 Resource structure and HTTP methods

Figures 9.3-1, 9.3-2, 9.3-3 and 9.3-4 show resource structures of the Redfish® API [2] resources that are applicable for the physical resource performance management service interface.

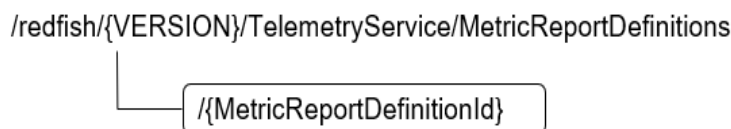


Figure 9.3-1: API resource structure of PmJob resource for the physical resource performance management service interface

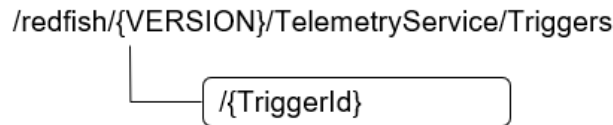


Figure 9.3-2: API resource structure of Threshold resource for the physical resource performance management service interface

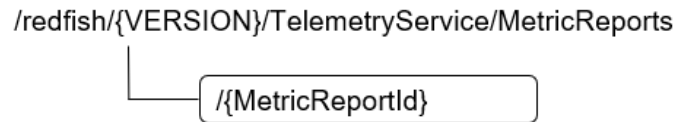


Figure 9.3-3: API resource structure of PerformanceReport resource for the physical resource performance management service interface



Figure 9.3-4: API resource structure of Subscription resource for physical resource performance management service interface

Table 9.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 9.3-1 shall be supported by the PIM. The HTTP methods marked as "O" (optional) in the "Cat" column may be optionally supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition. The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 9.3-1 shall be supported by the PIM. The HTTP methods marked as "O" (optional) in the "Cat" column may be optionally supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition.

Table 9.3-1: API Resources and HTTP methods of physical resource performance management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
PmJobs	/redfish/v1/TelemetryService/MetricReportDefinitions	GET	M	Get a collection of PmJobs.
		POST	M	Create a new "IndividualPmJob" resource.
IndividualPmJob	/redfish/v1/TelemetryService/MetricReportDefinitions/{MetricReportDefinitionId}	GET	M	Get information about the "IndividualPmJob" resource
		PATCH	M	Modify information of an "IndividualPmJob" resource.
		PUT	O	Replace an "IndividualPmJob" resource.
		DELETE	M	Delete an "IndividualPmJob" resource.
Thresholds	/redfish/v1/TelemetryService/Triggers	GET	M	Get a collection of Threshold resource.
		POST	M	Create a new "Individual Threshold" resource.
Individual Threshold	/redfish/v1/TelemetryService/Triggers/{TriggerId}	GET	M	Get information about an Individual Threshold" resource.
		PATCH	M	Modify an Individual Threshold" resource.
		PUT	O	Replace an Individual Threshold" resource.
		DELETE	M	Delete an Individual Threshold" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Performance Reports	/redfish/v1/TelemetryService/MetricReports	GET	M	Retrieve a collection of PerformancReports.
Individual Performance Report	/redfish/v1/TelemetryService/MetricReports/{MetricReportId}	GET	M	Retrieve an individual PerformancReport resource.
Subscriptions	/redfish/v1/EventService/Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 1). Can also be used to send a performance report asynchronously (see note 2). a subscribed event (see note 1). Can also be used to send a performance report asynchronously (see note 2).
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 3)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer.				
NOTE 2: See clause 9.4.8 for the flow related to the API consumers receiving performance reports asynchronously as notifications.				
NOTE 3: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

9.4 Sequence diagrams (informative)

9.4.1 Introduction

The sequence diagrams provided in the subsequent sub-clauses are generalized to indicate management of a generic Performance Management (PM) related object, which refers to either PmJob or Threshold specified in clause 5 of the present document. These diagrams and their description contain placeholders such as <PM object collection URI>, <PM object URI> and <PM object>, which correspond to the appropriate resource URIs and PM-related objects specified in clauses 9.3 and 5.4 respectively.

Sequence diagrams related to retrieval of performance reports do not contain generic placeholders. Specific resource URIs and object names are used in those sequence diagrams.

Flows related to task monitoring, subscription management and notifications are described in Annex A of the present document.

9.4.2 Creation of a PM object

This clause describes a sequence for creating an individual resource corresponding to a PM object.

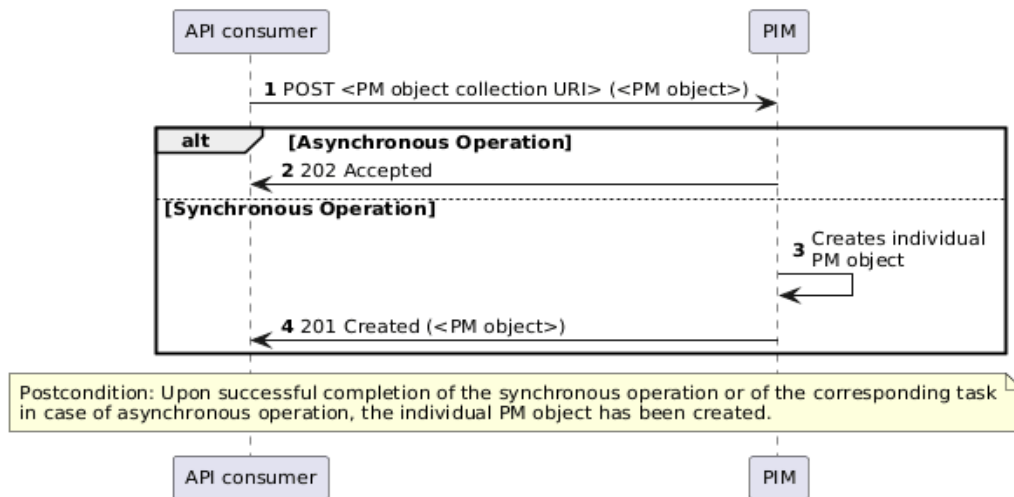


Figure 9.4.2-1: Flow of PM object creation

The creation of a PM object, as illustrated in Figure 9.4.2-1, consists of the following steps.

Precondition: None.

1. The API consumer sends a POST request to the <PM object collection URI> including a valid <PM object> representation in the message content body.
2. If the PIM supports asynchronous operations and creation of the PM object resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM creates an individual PM object.
4. The PIM returns a "201 Created" response to the API consumer and includes in the message content body a representation of the created PM object. The response also contains the URI of the newly created PM object in the Location header.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual PM object has been created.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.3 Querying information about PM objects

This clause describes the sequences for querying information about multiple and individual PM objects.

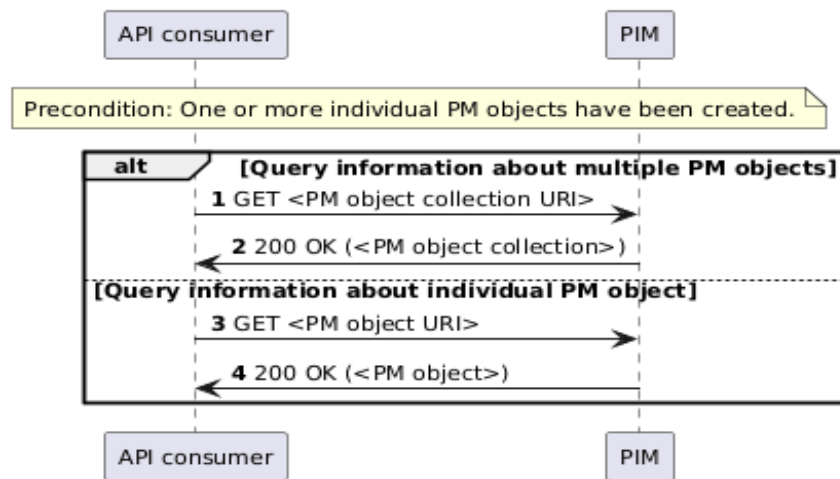


Figure 9.4.3-1: Flow of querying information about PM objects

The querying of information about one or more PM objects, as illustrated in Figure 9.4.3-1, consists of the following steps.

Precondition: One or more individual PM objects have previously been created.

1. If the API consumer intends to get information about multiple PM objects, it sends a GET request to the <PM object collection URI>.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a <PM object collection>, containing information about all the individual PM objects that have been created.
3. If the API consumer intends to get information about an individual PM object, it sends a GET request to the <PM object URI>.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual PM object.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.4 Modifying a PM object

This clause describes a sequence for modifying an individual PM object.

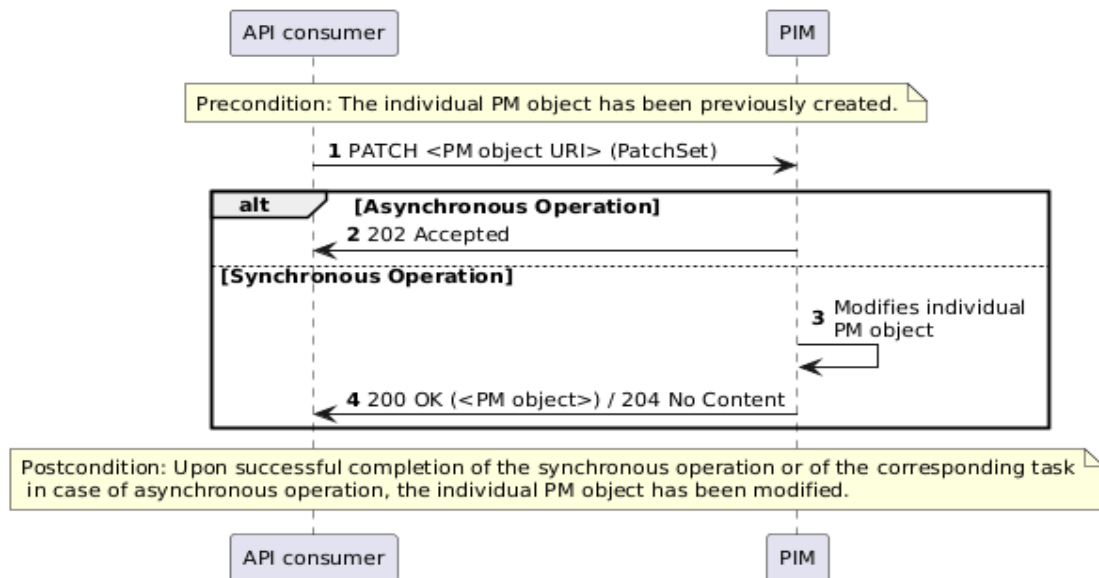


Figure 9.4.4-1: Flow of PM object modification

The modification of a PM object, as illustrated in Figure 9.4.4-1, consists of the following steps.

Precondition: The individual PM object has been previously created.

1. The API consumer sends a PATCH request to the <PM object URI>, including data structure in the message content body representing the modifiable properties to be modified in the individual PM object.
2. If the PIM supports asynchronous operations and modification of the PM object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM modifies the individual PM object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified PM object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual PM object has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.5 Replacing a PM object

This clause describes a sequence for replacing an individual PM object.

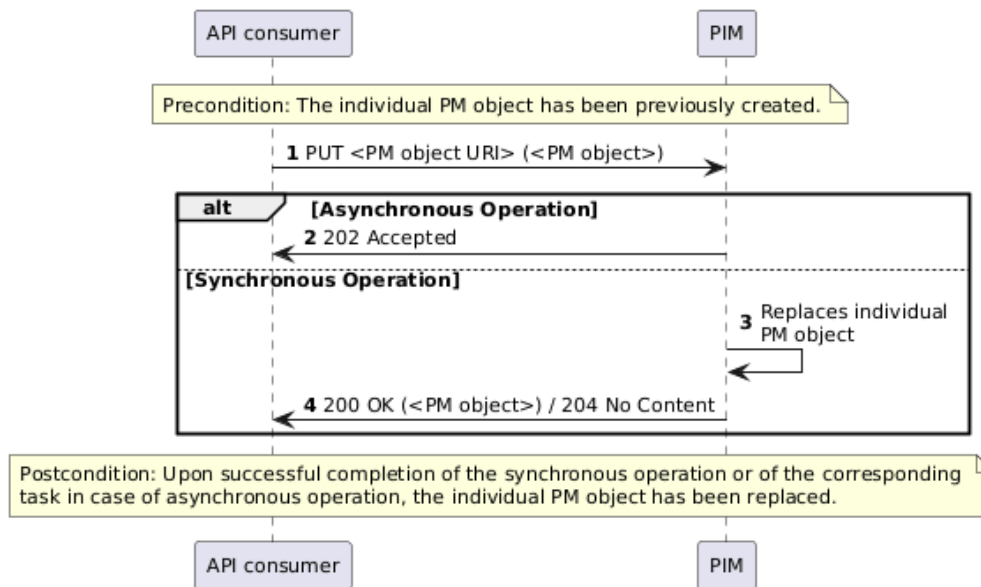


Figure 9.4.5-1: Flow of PM object replacement

The replacement of an individual PM object, as illustrated in Figure 9.4.5-1, consists of the following steps.

Precondition: The individual PM object has been previously created.

1. The API consumer sends a PUT request to the <PM object URI> including a valid representation of the individual PM object to be replaced in the message content body.
2. If the PIM supports asynchronous operations and replacement of the PM object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM replaces the individual PM object.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced PM object. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual PM object has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.6 Deleting a PM object

This clause describes a sequence for deleting an individual PM object.

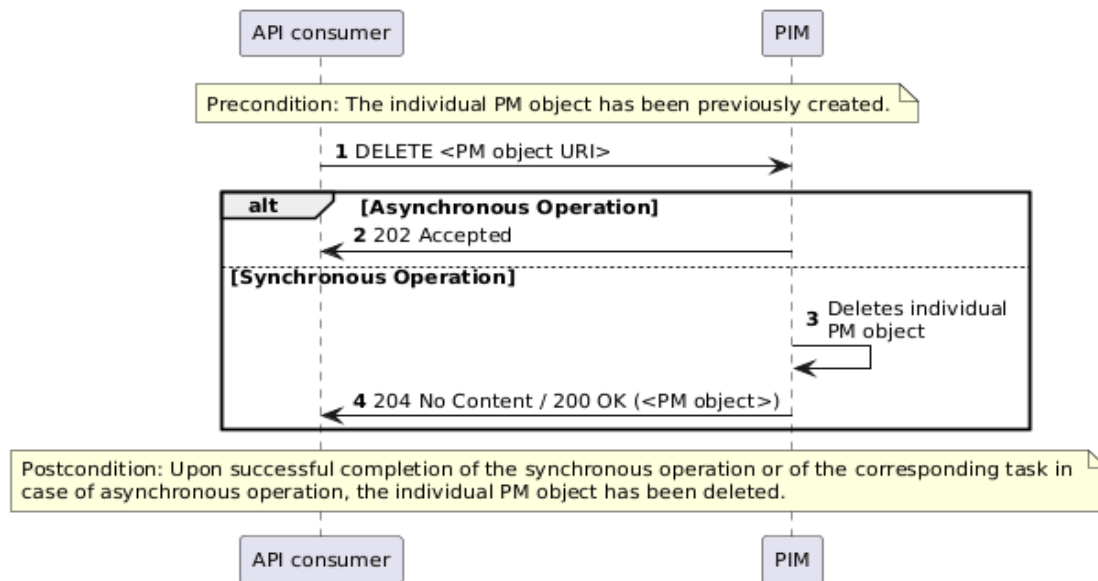


Figure 9.4.6-1: Flow of PM object deletion

The deletion of an individual PM object, as illustrated in Figure 9.4.6-1, consists of the following steps.

Precondition: The individual PM object has been previously created.

1. The API consumer sends a DELETE request to the <PM object URI>.
2. If the PIM supports asynchronous operations and deletion of the PM object requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM deletes the individual PM object.
4. The PIM returns a "204 No Content" response to the API consumer indicating that the PM object has been successfully deleted. Alternatively, the PIM can also return a "200 OK" response to the API consumer including a representation of the deleted PM object in the message content body.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual PM object has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.7 Retrieving performance reports

This clause describes the sequence for querying information about multiple and individual performance reports.

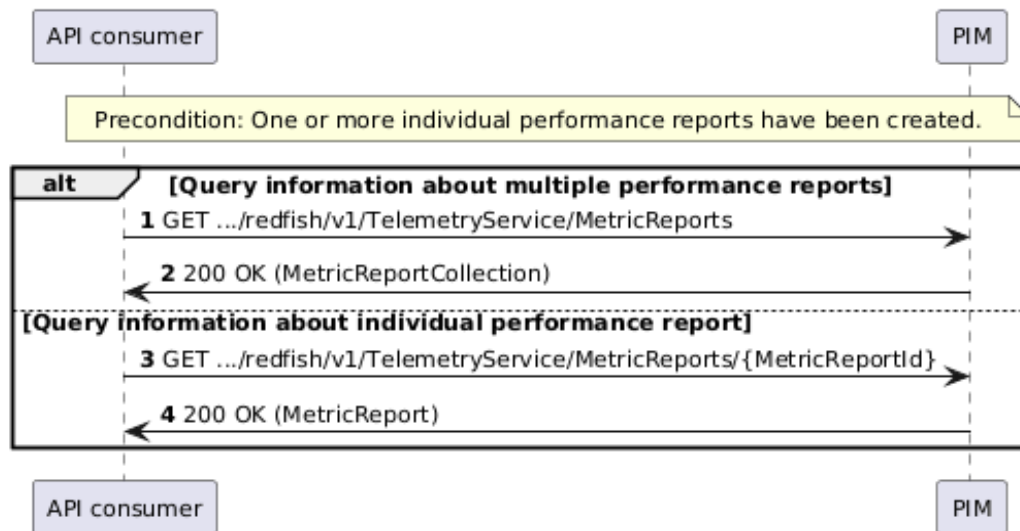


Figure 9.4.7-1: Flow of querying information about performance reports

The querying of information about one or more performance reports, as illustrated in Figure 9.4.7-1, consists of the following steps.

Precondition: One or more individual performance reports have previously been generated by the PIM.

1. If the API consumer intends to get information about multiple performance reports, it sends a GET request to the performance reports URI.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a collection of performance reports, containing information about all the individual performance reports that have been generated.
3. If the API consumer intends to get information about an individual performance report, it sends a GET request to the URI representing individual performance report.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual performance report.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

9.4.8 Receiving performance reports asynchronously

This clause describes the sequence for receiving performance reports asynchronously as notifications.

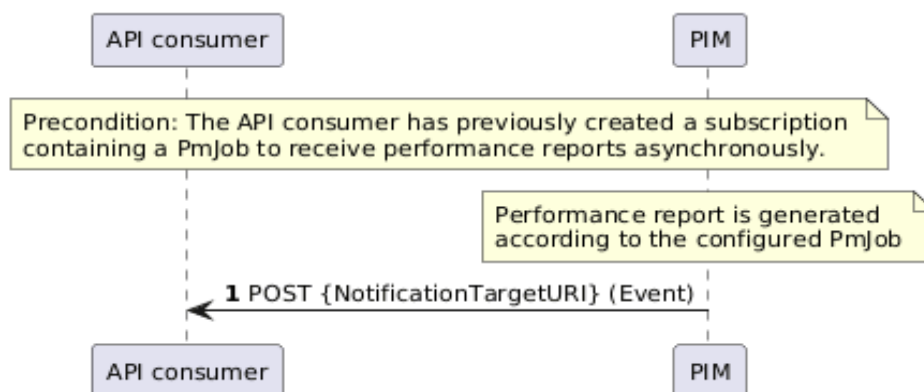


Figure 9.4.8-1: Flow of receiving performance reports as notifications

The procedure consists of the following steps as illustrated in Figure 9.4.8-1.

Precondition: The API consumer has previously created a subscription containing a PmJob to receive performance reports asynchronously.

1. When generating a performance report according to the configured PmJob (*MetricReportDefinition*) as part of the created subscription, an event of type *MetricReport* gets triggered. As a result of this, the PIM generates a notification that includes the performance report and sends it in the body of a POST request to the notification target URI which the API consumer had registered as part of the subscription request.

Postcondition: None.

Error handling: None.

9.5 Resources

9.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource performance management service interface.

9.5.2 Resource: PmJobs

This API resource represents the performance management jobs (PmJobs) created for reporting of performance metrics related to physical resources managed by the PIM. The API resource can be used to retrieve a collection of available PmJobs as well as to create an individual PmJob resource.

Table 9.5.2-1 specifies the supported HTTP methods for this API resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.2-1: PmJobs resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../MetricReportDefinitions (see note)	GET	Lists a collection of available PmJobs. Retrieves performance management jobs created for collecting metrics related to physical resources managed by the PIM.	PimPrPmMgt.003
	POST	Creates a new "Individual PmJob" resource.	PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.3 Resource: Individual PmJob

This API resource represents an individual PmJob created for reporting of performance metrics related to physical resources managed by the PIM. The API resource can be used to query, update or delete an individual PmJob resource.

Table 9.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.3-1: Individual PmJob resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../MetricReportDefinitions/{MetricReportDefinitionId} (see note)	GET	Retrieves information about an individual PmJob managed by the PIM.	PimPrPmMgt.003
	PUT	Modifies the individual PmJob resource by replacing it whole.	PimPrPmMgt.003
	PATCH	Modifies the individual PmJob resource by updating only a patch of it.	PimPrPmMgt.003
	DELETE	Deletes the individual PmJob resource.	PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.4 Resource: Thresholds

This API resource represents thresholds and triggers for metrics related to physical resources managed by the PIM. The API resource can be used to retrieve a collection of available thresholds as well as to create an individual threshold.

Table 9.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.4-1: Thresholds resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Triggers (see note)	GET	Lists available thresholds.	PimPrPmMgt.003
	POST	Creates a new threshold.	PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.5 Resource: Individual Threshold

This resource represents an individual threshold for metrics related to physical resources managed by the PIM. The API resource can be used to query, update or delete an individual threshold resource.

Table 9.5.5-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.5-1: Individual Threshold resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Triggers/{TriggerId}	GET	Retrieves an individual threshold.	PimPrPmMgt.003
	PUT	Modifies the individual threshold resource by replacing it whole.	PimPrPmMgt.003
	PATCH	Modifies the individual threshold resource by updating only a patch of it.	PimPrPmMgt.003
	DELETE	Deletes the individual threshold resource.	PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.6 Resource: Performance Reports

This API resource represents performance reports generated by the PIM based on the PmJobs related to metric collection of physical resources. The API resource can be used to retrieve multiple performance reports that have been generated by the PIM.

Table 9.5.6-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.6-1: Performance Reports resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../MetricReports (see note)	GET	Retrieves a list of performance reports generated by the PIM.	PimPrPmMgt.001
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.7 Resource: Individual Performance Report

This API resource represents an individual performance report generated by the PIM related to metric collection of physical resources and can be used to retrieve an individual performance report.

Table 9.5.7-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.7-1: Individual Performance Report resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../MetricReports/{MetricReportId} (see note)	GET	Retrieves an individual performance report generated by the PIM.	PimPrPmMgt.001
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.8 Resource: Subscriptions

This resource represents subscriptions related to performance management of physical resources and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource. This resource can also be used to create a PmJob in the form of a subscription for receiving performance reports asynchronously as notifications.

Table 9.5.8-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 9.5.8-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of subscriptions. Retrieves information about the created subscriptions related to performance management of physical resources managed by the PIM.	PimPrPmMgt.004
	POST	Creates a new Individual Subscription resource. A subscription representing a PmJob can be created for retrieving performance reports asynchronously as notifications.	PimPrPmMgt.004 PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.9 Resource: Individual Subscription

This resource represents an individual subscription related to performance management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resource.

Table 9.5.9-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 9.5.9-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription resource managed by the PIM.	PimPrPmMgt.004
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrPmMgt.004
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrPmMgt.004
	DELETE	Deletes the individual subscription resource.	PimPrPmMgt.004
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.5.10 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 9.5.10-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 9.5.10-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource performance management once that operation has been completed.	PimPrPmMgt.003
NOTE: Refer to Table 9.3-1 for the full URI of this resource.			

9.6 Data model

The request and response data structures of the physical resource performance management service interface are defined in the Redfish® data model specification [3].

10 Physical resource fault management service interface

10.1 Description

The physical resource fault management service interface allows the API consumers to invoke fault management operations related to physical resources. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Receive information (via notifications) about faults and alarms associated with physical resources.
- Subscribe to notifications related to faults and alarms.
- Notify subscribers according to the subscriptions.

NOTE: In Redfish®, event information is delivered to subscribed consumers via notifications rather than through direct retrieval. Information about previously generated events (including alarms) may be retrieved with same identifier of events via Redfish log services, while supported message types can be discovered via message registries [2].

10.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1/EventService/Subscriptions

10.3 Resource structure and HTTP methods

Figure 10.3-1 shows the resource structure of the Redfish® API [2] resources that are applicable for the physical resource fault management service interface.



Figure 10.3-1: API resource structure of Subscription resource for physical resource fault management service interface

Table 10.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 10.3-1 shall be supported by the PIM. The HTTP methods marked as "O" (optional) in the "Cat" column may be optionally supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition.

Table 10.3-1: API Resources and HTTP methods of physical resource fault management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Subscriptions	/redfish/v1/EventService/Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 1). This resource is used to send information about faults to the consumer in the form of an alert type event.
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 2)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer.				
NOTE 2: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

10.4 Sequence diagrams (informative)

10.4.1 Introduction

The physical resource fault management service interface supports notifying the subscribed consumers about any faults or alarms that may occur related to the concerned physical resources. Flows related to subscription management and notifications described in Annex A of the present document are also applicable for the subscriptions and notifications related to fault management supported by this service interface.

10.5 Resources

10.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource fault management service interface.

10.5.2 Resource: Subscriptions

This resource represents subscriptions related to fault management of physical resources and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource. This resource can be used to create a subscription for receiving notifications about faults associated with physical resources.

Table 10.5.2-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 10.5.2-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of subscriptions. Retrieves information about the created subscriptions related to fault management of physical resources managed by the PIM.	PimPrFmMgt.003
	POST	Creates a new Individual Subscription resource.	PimPrFmMgt.003
NOTE: Refer to Table 10.3-1 for the full URI of this resource.			

10.5.3 Resource: Individual Subscription

This resource represents an individual subscription related to fault management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resource.

Table 10.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 10.5.3-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription resource managed by the PIM.	PimPrFmMgt.003
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrFmMgt.003
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrFmMgt.003
	DELETE	Deletes the individual subscription resource.	PimPrFmMgt.003
NOTE: Refer to Table 10.3-1 for the full URI of this resource.			

10.5.4 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 10.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 10.5.4-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource log management once that operation has been completed.	PimPrFmMgt.003
NOTE: Refer to Table 10.3-1 for the full URI of this resource.			

10.6 Data model

The request and response data structures of the physical resource fault management service interface are defined in the Redfish® data model specification [3].

11 Physical resource log management service interface

11.1 Description

The physical resource log management service interface allows the API consumers to invoke log management operations related to physical resources. The Redfish® interface [2] endpoints profiled for this service interface allow consumers to manage API resources that are applicable for this service interface.

The operations supported over this service interface are the following:

- Create, update, retrieve and delete Logging Jobs.
- Retrieve information about Logs collected by Logging Jobs.
- Subscribe to notifications related to log collection events.
- Notify subscribers according to the subscriptions.

11.2 API version

The API {VERSION} for the profiled Redfish® interface [2] shall be set to "v1". Details about the API structure of the Redfish® interface [2] are provided in clause 4.3.2 of the present document.

The corresponding Redfish® API roots are specified below:

- /redfish/v1/.../LogServices
- /redfish/v1/EventService/Subscriptions

NOTE: The Redfish® LogService can be configured to collect logs for different kinds of Redfish® resources. The resource URI for the corresponding LogService varies accordingly. For example, the resource URI of a logging job configured for an individual ComputerSystem resource may look like /redfish/v1/Systems/{ComputerSystemId}/LogServices/{LogServiceId}. Therefore, the API root for the LogService listed above is kept generic on purpose. Full list of possible URI paths for the LogService are specified in the Redfish® data model specification [3].

11.3 Resource structure and HTTP methods

Figures 11.3-1 and 11.3-2 show resource structures of the Redfish® API [2] resources that are applicable for the physical resource log management service interface.

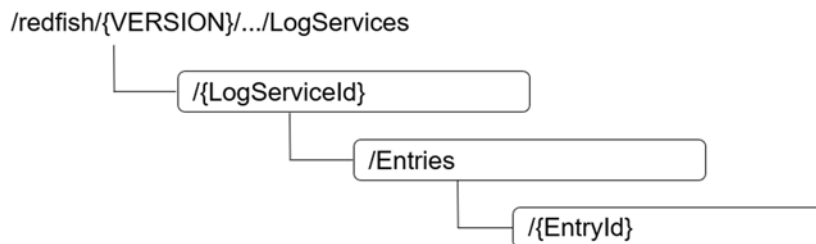


Figure 11.3-1: API resource structure of LoggingJob and Log resources for the physical resource log management service interface



Figure 11.3-2: API resource structure of Subscription resource for physical resource log management service interface

Table 11.3-1 lists the API resources, and the applicable HTTP methods for each resource endpoint.

The HTTP methods marked as "M" (mandatory) in the "Cat" column of Table 11.3-1 shall be supported by the PIM. The HTTP methods marked as "O" (optional) in the "Cat" column may be optionally supported by the PIM. The HTTP methods marked as "CM" (conditional mandatory) shall be supported by the PIM depending on the corresponding condition.

Table 11.3-1: API Resources and HTTP methods of physical resource log management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
LoggingJobs	/redfish/v1/.../LogServices (see note 1)	GET	M	Get a collection of LoggingJobs.
		POST	M	Create a new "IndividualLoggingJob" resource.
IndividualLoggingJob	/redfish/v1/.../LogServices /{LogServiceId} (see note 1)	GET	M	Get an "IndividualLoggingJob" resource.
		PATCH	M	Modify an "IndividualLoggingJob" resource.
		PUT	O	Replace an "IndividualLoggingJob" resource.
		DELETE	M	Delete an "IndividualLoggingJob" resource.
Logs	/redfish/v1/.../LogServices /{LogServiceId}/Entries (see note 1)	GET	M	Get a collection of Logs.
IndividualLog	/redfish/v1/.../LogServices /{LogServiceId}/Entries /{EntryId} (see note 1)	GET	M	Get information about an "IndividualLog" resource.
Subscriptions	/redfish/v1/EventService /Subscriptions	POST	M	Create a new "Individual Subscription" resource.
		GET	M	Get the collection of all available subscriptions.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual Subscription	/redfish/v1/EventService/Subscriptions/{SubscriptionId}	GET	M	Get information about the "Individual Subscription" resource.
		PATCH	M	Modify an "Individual Subscription" resource.
		PUT	O	Replace an "Individual Subscription" resource.
		DELETE	M	Delete an "Individual Subscription" resource.
Notification endpoint	{NotificationTargetURI}	POST	M	Send a notification related to a subscribed event (see note 2). This resource can also be used to notify the consumer about availability of logs. This resource is also used to send logs asynchronously to the consumer if an event triggers a log entry.
Individual Task Monitor	/redfish/v1/TaskService/TaskMonitors/{TaskMonitorId} (see note 3)	GET	CM	Enable API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed. This resource endpoint shall be supported if the API producer supports asynchronous operations.
NOTE 1: Possible URI paths for the LogService and LogEntry resources are specified in the Redfish® data model specification [3]. NOTE 2: Notification endpoint resource and the corresponding NotificationTargetURI is used by the API producer to send notifications to the API consumer. This resource is not produced by the API producer. NOTE 3: The URI of the task monitor is provided in the Location header of the HTTP response to the API consumer when the API producer returns a "202 Accepted" response.				

11.4 Sequence diagrams (informative)

11.4.1 Introduction

Flows related to task monitoring, subscription management and notifications are described in Annex A of the present document.

11.4.2 Creation of a Logging Job

This clause describes a sequence for creating an individual Logging Job.

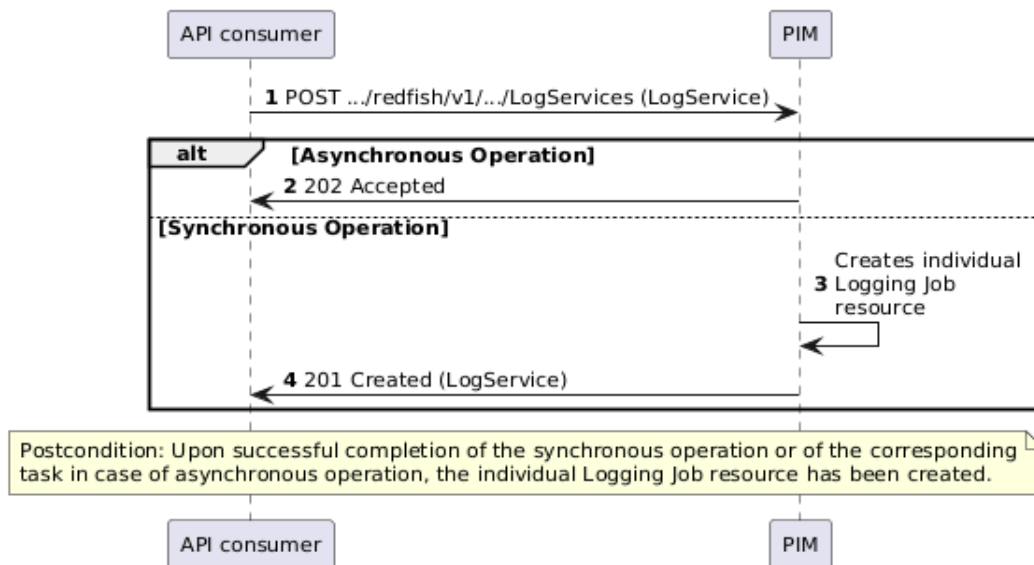


Figure 11.4.2-1: Flow of Logging Job creation

The creation of a Logging Job object, as illustrated in Figure 11.4.2-1, consists of the following steps.

Precondition: None.

1. The API consumer sends a POST request to the LogService resource URI including a valid representation of the individual Logging Job resource in the message content body.
2. If the PIM supports asynchronous operations and creation of the Logging Job resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM creates an individual Logging Job resource.
4. The PIM returns a "201 Created" response to the API consumer and includes in the message content body a representation of the created Logging Job resource. The response also contains the URI of the newly created resource in the Location header.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the individual Logging Job resource has been created.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.3 Querying information about Logging Job resources

This clause describes the sequences for querying information about both Logging Job resource collection and individual Logging Job resource.

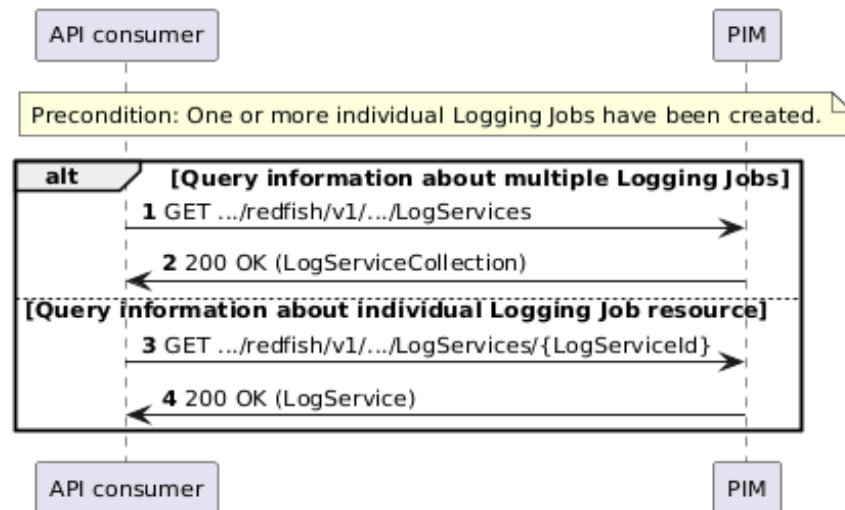


Figure 11.4.3-1: Flow of querying information about Logging Job resources

The querying of information about one or more Logging Job resources, as illustrated in Figure 11.4.3-1, consists of the following steps.

Precondition: One or more individual Logging Job resources have been created.

1. If the API consumer intends to get information about multiple Logging Job resources, it sends a GET request to the LogService resource URI.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a Logging Job collection, containing information about all the individual Logging Job resources that have been created.
3. If the API consumer intends to get information about an individual Logging Job resource, it sends a GET request to the Individual Logging Job resource identified by its name in the URI.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual Logging Job resource.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.4 Modifying a Logging Job resource

This clause describes a sequence for modifying an individual Logging Job resource.

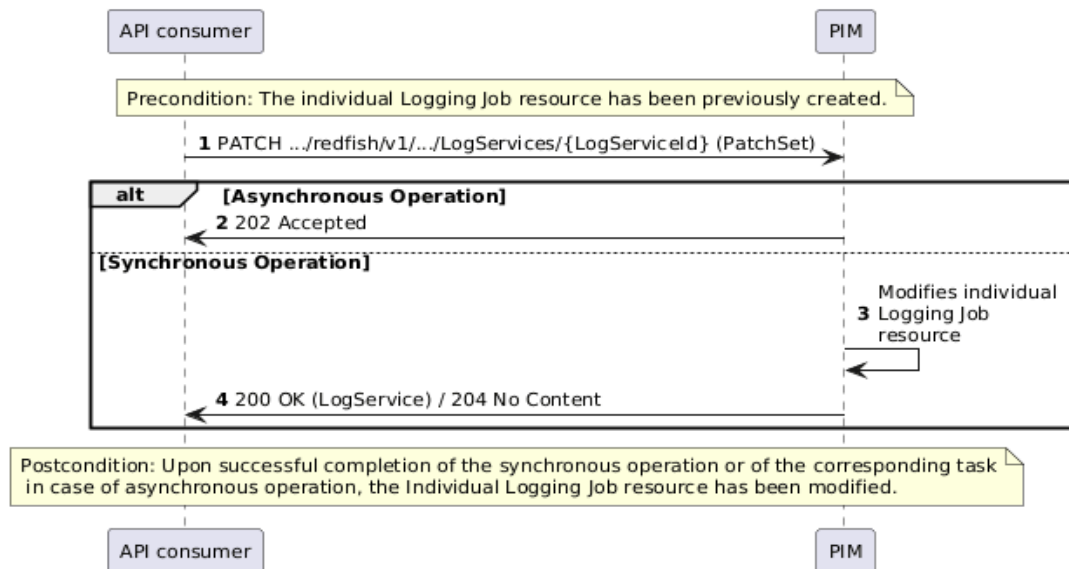


Figure 11.4.4-1: Flow of Logging Job modification

The modification of an individual Logging Job resource, as illustrated in Figure 11.4.4-1, consists of the following steps.

Precondition: The individual Logging Job resource has been previously created.

1. The API consumer sends a PATCH request to the Individual Logging Job resource, including data structure in the message content body representing the modifiable properties to be modified in the individual Logging Job resource.
2. If the PIM supports asynchronous operations and modification of the Logging Job resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM modifies the Individual Logging Job resource.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified Logging Job resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Logging Job resource has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.5 Replacing a Logging Job resource

This clause describes a sequence for replacing an individual Logging Job resource.

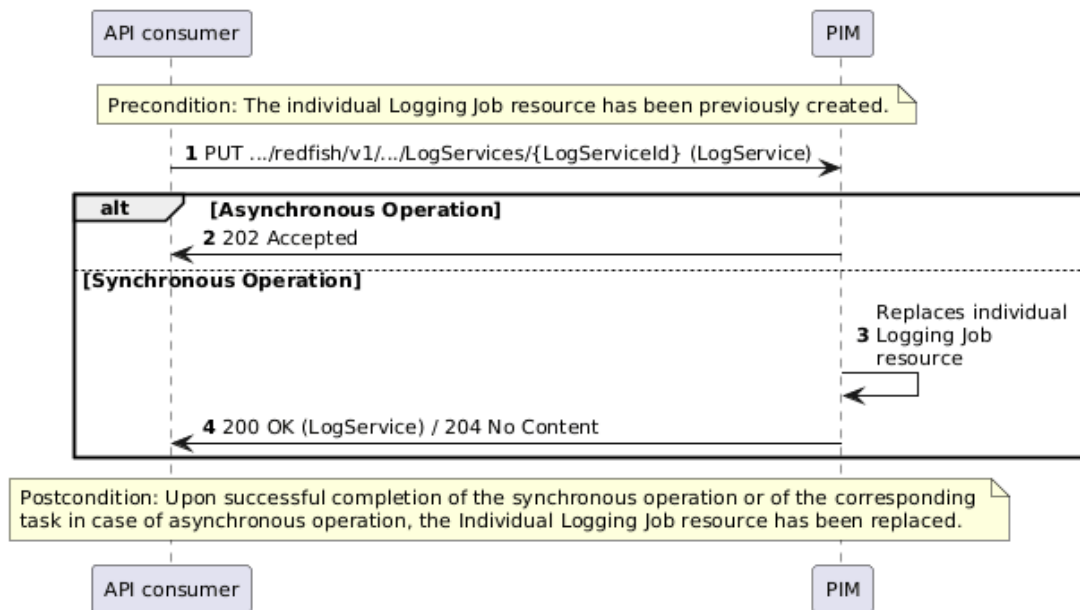


Figure 11.4.5-1: Flow of Logging Job replacement

The replacement of an individual Logging Job resource, as illustrated in Figure 11.4.5-1, consists of the following steps.

Precondition: The individual Logging Job resource has been previously created.

1. The API consumer sends a PUT request to the Individual Logging Job resource including a valid representation of the individual Logging Job resource to be replaced in the message content body.
2. If the PIM supports asynchronous operations and replacement of the Logging Job resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM replaces the Individual Logging Job resource.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced Logging Job resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Logging Job resource has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.6 Deleting a Logging Job resource

This clause describes a sequence for deleting an Individual Logging Job resource.

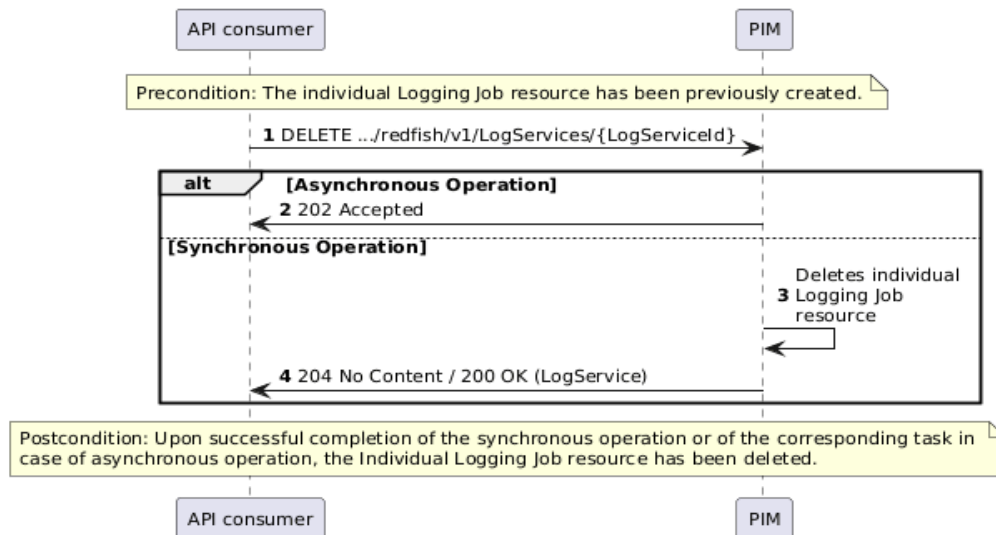


Figure 11.4.6-1: Flow of Logging Job deletion

The deletion of an individual Logging Job resource, as illustrated in Figure 11.4.6-1, consists of the following steps.

Precondition: The individual Logging Job resource has been previously created.

1. The API consumer sends a DELETE request to the Individual Logging Job resource.
2. If the PIM supports asynchronous operations and deletion of the Logging Job resource requires some time to process, the PIM returns a "202 Accepted" response. The response also contains the URI of the associated Task Monitor in the Location header. The response body can optionally contain representation of the corresponding Task resource created by the PIM for this asynchronous operation.
3. The PIM deletes the individual Logging Job resource.
4. The PIM returns a "204 No Content" response to the API consumer indicating that the resource has been successfully deleted. Alternatively, the PIM can also return a "200 OK" response to the API consumer including a representation of the deleted Logging Job resource in the message content body.

Postcondition: Upon successful completion of the synchronous operation or of the corresponding task in case of asynchronous operation, the Individual Logging Job resource has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.7 Querying information about logs

This clause describes the sequences for querying information about multiple log entries and individual log entry.

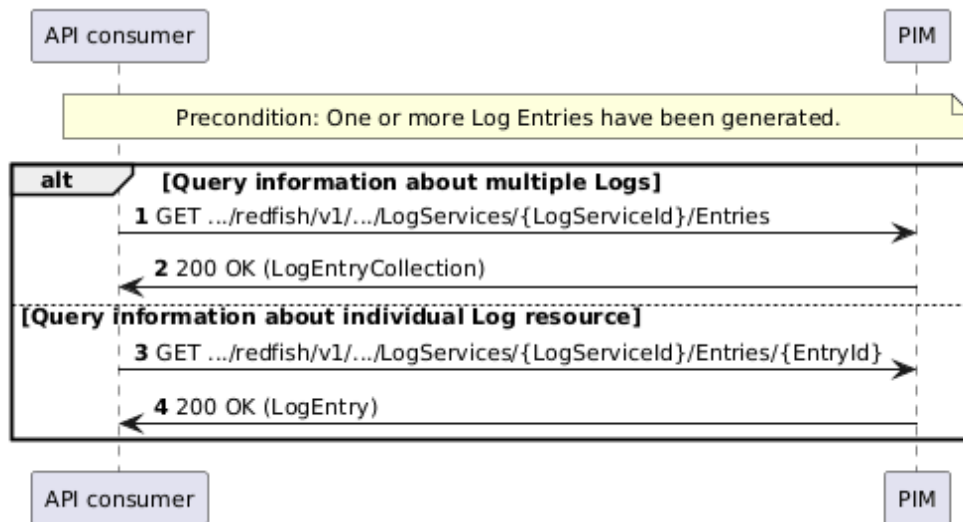


Figure 11.4.7-1: Flow of querying information about Logs

The querying of information about one or more Log entries, as illustrated in Figure 11.4.7-1, consists of the following steps.

Precondition: One or more Log Entries have been generated by the PIM.

1. If the API consumer intends to get information about multiple Log entries, it sends a GET request to the Logs collection URI.
2. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a Logs collection, containing information about all the individual log entries that have been collected.
3. If the API consumer intends to get information about an individual log entry, it sends a GET request to the Individual Log resource identified by its name in the URI.
4. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual Log entry.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.8 Receiving diagnostics related logs

This clause describes the sequence for receiving diagnostics related logs collected by the PIM.

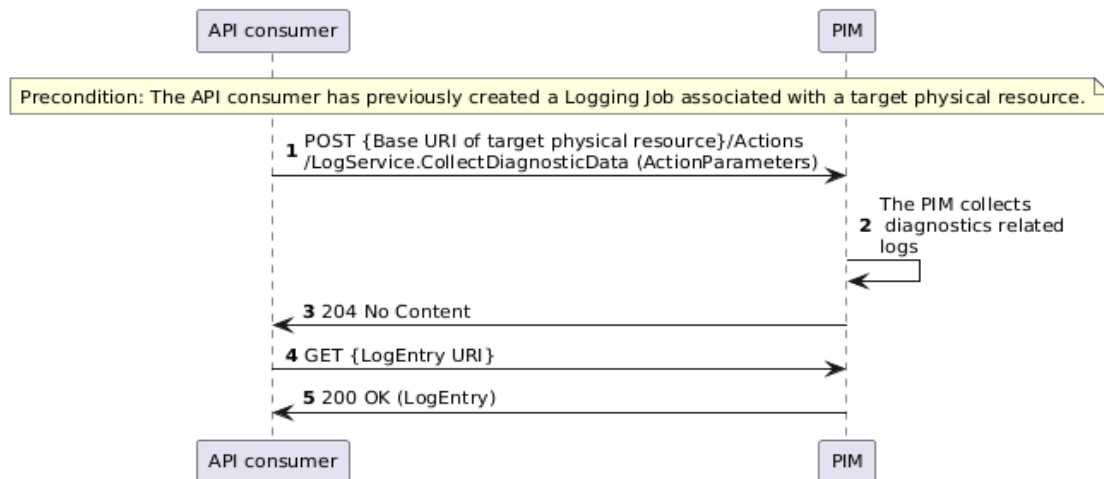


Figure 11.4.8-1: Flow of receiving diagnostics related logs

The procedure of receiving logs related to diagnostics consists of the following steps as illustrated in Figure 11.4.8-1.

Precondition: The API consumer has previously created a Logging Job associated with a target physical resource.

1. The API consumer sends a POST request to {Base URI of target physical resource}/Actions/LogService.CollectDiagnosticData and includes the necessary action parameters in the message body.
2. The PIM collects the logs to generate the diagnostics report.
3. Upon successful completion of the action, the PIM returns a "204 No Content" response to the API consumer indicating that the action has been performed successfully. The Location header in the response contains a URI of a resource, which has the type LogEntry and contains the collected diagnostic data.
4. The API consumer sends a GET request to the LogEntry resource URI provided by the PIM in the Location Header of the "204 No Content" response.
5. The PIM responds with "200 OK" and valid representation of LogEntry resource containing the logs.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

11.4.9 Receiving notification about availability of diagnostics related logs

This clause describes the sequence for receiving notification about availability of diagnostics related logs.

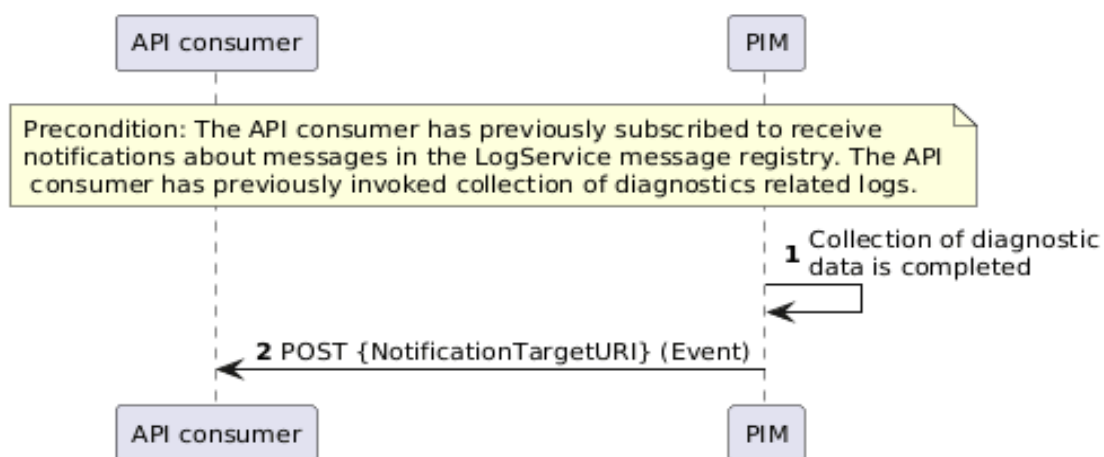


Figure 11.4.9-1: Flow of receiving notification about availability of diagnostics related logs

The procedure consists of the following steps as illustrated in Figure 11.4.9-1.

Precondition: The API consumer has previously subscribed to receive notifications about messages in the LogService message registry. The API consumer has previously invoked collection of diagnostics related logs.

1. An event is triggered once the collection of diagnostic data is completed by the PIM, resulting in generation of a new DiagnosticDataCollected message in the LogService message registry.
2. The PIM sends the DiagnosticDataCollected message as an event notification to the consumer indicating availability of diagnostics related logs.

Postcondition: None.

Error handling: None.

11.5 Resources

11.5.1 Introduction

The present clause specifies API resources (endpoints) of the physical resource log management service interface.

11.5.2 Resource: LoggingJobs

This API resource represents the logging jobs (LoggingJobs) created for collecting logs related to physical resources managed by the PIM. The API resource can be used to retrieve a collection of available LoggingJobs as well as to create an individual LoggingJob resource.

Table 11.5.2-1 specifies the supported HTTP methods for this API resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 11.5.2-1: LoggingJobs resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
/redfish/v1/.../LogServices (see note)	GET	Lists a collection of available LoggingJobs. Retrieves logging management jobs created for collecting logs related to physical resources managed by the PIM.	PimPrLmMgt.003
	POST	Creates a new "Individual LoggingJob" resource.	PimPrLmMgt.003
NOTE: Possible URI paths for the LogService resource are specified in the Redfish® data model specification [3].			

11.5.3 Resource: Individual LoggingJob

This API resource represents an individual LoggingJob created for collecting of logs related to physical resources managed by the PIM. The API resource can be used to query, update or delete an individual LoggingJob resource.

Table 11.5.3-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 11.5.3-1: Individual LoggingJob resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
/redfish/v1/.../LogServices/{LogServiceId} (see note)	GET	Retrieves information about an individual LoggingJob managed by the PIM.	PimPrLmMgt.003
	PUT	Modifies the individual LoggingJob resource by replacing it whole.	PimPrLmMgt.003
	PATCH	Modifies the individual LoggingJob resource by updating only a patch of it.	PimPrLmMgt.003
	DELETE	Deletes the individual LoggingJob resource.	PimPrLmMgt.003
NOTE: Possible URI paths for the LogService resource are specified in the Redfish® data model specification [3].			

11.5.4 Resource: Logs

This API resource represents logs collected by the PIM based on the LoggingJobs related to log collection of physical resources. The API resource can be used to retrieve multiple log entries that have been generated by the PIM.

Table 11.5.4-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 11.5.4-1: Logs resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
/redfish/v1/.../LogServices/{LogServiceId}/Entries (see note)	GET	Retrieves a list of log entries generated by the PIM for a LoggingJob.	PimPrLmMgt.004
NOTE: Possible URI paths for the LogService resource are specified in the Redfish® data model specification [3].			

11.5.5 Resource: Individual Log

This API resource represents an individual log entry generated by the PIM against a LoggingJob created for collection of logs related to physical resources managed by the PIM. The API resource can be used to retrieve an individual log entry.

Table 11.5.5-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish® specification [2].

Table 11.5.5-1: Individual Log resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
/redfish/v1/.../LogServices/{LogServiceId}/Entries/{EntryId} (see note)	GET	Retrieves a single log entry generated by the PIM for a LoggingJob.	PimPrLmMgt.004
NOTE: Possible URI paths for the LogService resource are specified in the Redfish® data model specification [3].			

11.5.6 Resource: Subscriptions

This resource represents subscriptions related to log management of physical resources and can be used to retrieve a collection of already created subscription resources as well as to create a new individual subscription resource. This resource can be used to create a subscription to receive a notification once the Redfish[®] service has successfully collected logs (diagnostics data) in the respective message registry.

Table 11.5.6-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 11.5.6-1: Subscriptions resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions (see note)	GET	Lists a collection of subscriptions. Retrieves information about the created subscriptions related to log management of physical resources managed by the PIM.	PimPrLmMgt.002
	POST	Creates a new Individual Subscription resource.	PimPrLmMgt.002
NOTE: Refer to Table 11.3-1 for the full URI of this resource.			

11.5.7 Resource: Individual Subscription

This resource represents an individual subscription related to log management of physical resources managed by the PIM, and can be used to query, update or delete the individual subscription resource.

Table 11.5.7-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 11.5.7-1: Individual Subscription resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../Subscriptions/{SubscriptionId} (see note)	GET	Retrieves information about an individual subscription resource managed by the PIM.	PimPrLmMgt.002
	PUT	Modifies the individual subscription resource by replacing it whole.	PimPrLmMgt.002
	PATCH	Modifies the individual subscription resource by updating only a patch of it.	PimPrLmMgt.002
	DELETE	Deletes the individual subscription resource.	PimPrLmMgt.002
NOTE: Refer to Table 11.3-1 for the full URI of this resource.			

11.5.8 Resource: Individual Task Monitor

This resource represents an individual task monitor for an asynchronous operation being performed by the PIM and can be used by the API consumer to monitor the progress of an asynchronous operation as well as receive the result of the requested operation once the corresponding task has been completed.

Table 11.5.8-1 specifies the supported HTTP methods for this resource and maps them against the PIM service interface requirements as specified in ETSI GS NFV-IFA 053 [1].

The URI query parameters, request and response bodies, and response codes of the individual resource methods shall follow the provisions specified in Redfish[®] specification [2].

Table 11.5.8-1: Individual Task Monitor resource methods mapped against PIM service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 053 [1]
.../TaskService/TaskMonitors/{TaskMonitorId} (see note)	GET	Retrieves result of an asynchronous operation related to physical resource log management once that operation has been completed.	PimPrLmMgt.002 PimPmLmMgt.003
NOTE: Refer to Table 11.3-1 for the full URI of this resource.			

11.6 Data model

The request and response data structures of the physical resource log management service interface are defined in the Redfish[®] data model specification [3].

Annex A (informative): Sequence Diagrams

A.1 Overview

The present clause provides sequence diagrams for the sequences that are common among different PIM service interfaces. Sequences related to the following aspects are covered in the present clause:

- for subscriptions management using the relevant Redfish[®] service, i.e. Event Service;
- receiving notifications from the Event Service; and
- task monitoring using Redfish[®] Task Service in case of asynchronous operations.

A.2 Flow of managing subscriptions

This clause describes the procedure for creating, querying/reading and deleting subscriptions to event notifications.

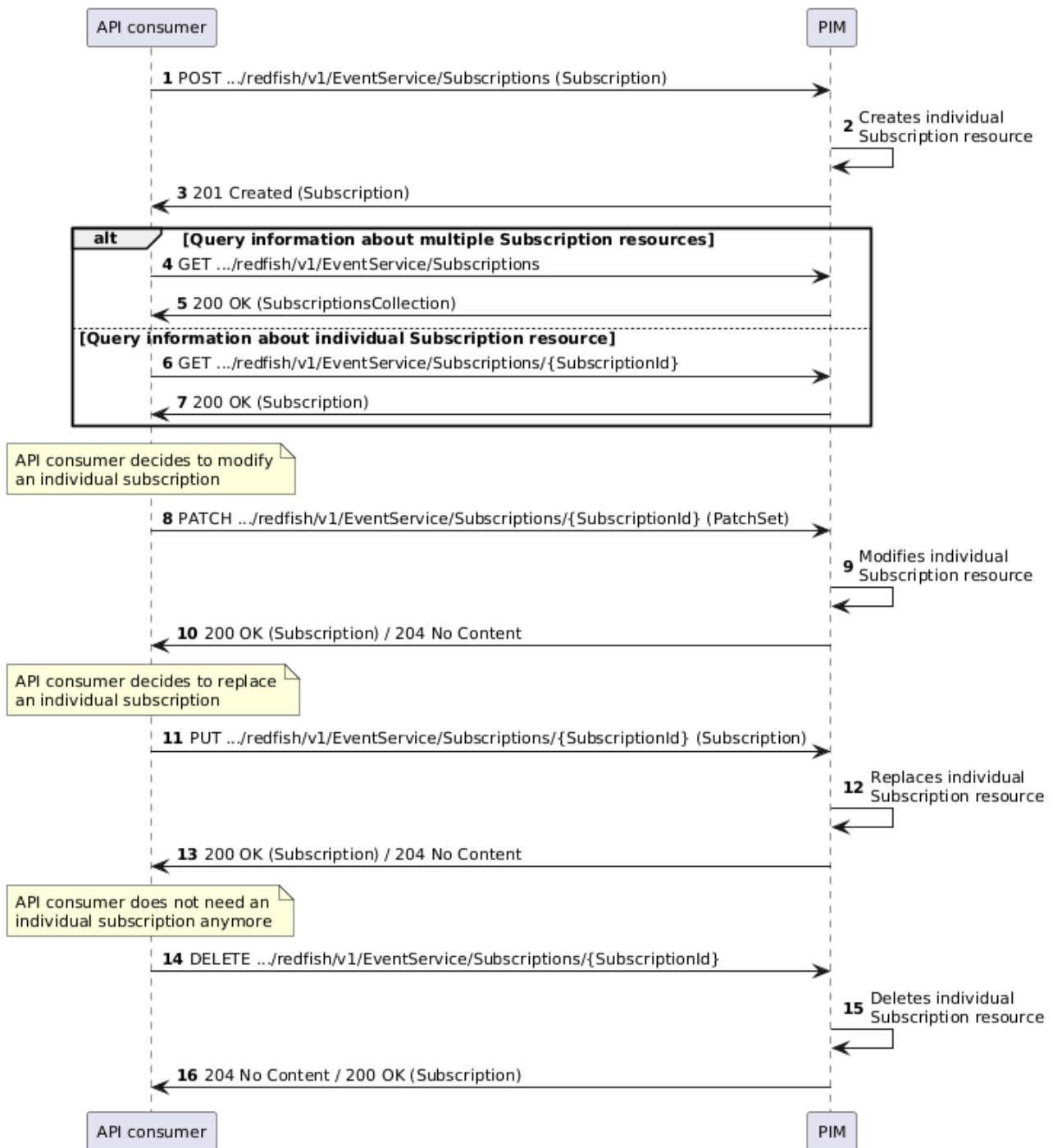


Figure A.2-1: Flow of managing subscriptions

The procedure consists of the following steps as illustrated in Figure A.2-1:

1. The API consumer sends a POST request to the Subscriptions resource including a valid representation of the individual Subscription resource in the message content body.
2. The PIM creates an individual Subscription resource.
3. The PIM returns a "201 Created" response to the API consumer and includes in the message content body a representation of the created Subscription resource. The response also contains the URI of the newly created resource in the Location header.
4. If the API consumer intends to get information about multiple Subscription resources, it sends a GET request to the Subscriptions resource.

5. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of a Subscriptions collection, containing information about all the individual Subscription resources that have been created.
6. If the API consumer intends to get information about an individual Subscription resource, it sends a GET request to the Individual Subscription resource identified by its name in the URI.
7. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the individual Subscription resource.
8. The API consumer sends a PATCH request to the Individual Subscription resource, including data structure in the message content body representing the modifiable properties to be modified in the Individual Subscription resource.
9. The PIM modifies the Individual Subscription resource.
10. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the modified Subscription resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.
11. The API consumer sends a PUT request to the Individual Subscription resource including a valid representation of the individual Subscription resource to be replaced in the message content body.
12. The PIM replaces the Individual Subscription resource.
13. The PIM returns a "200 OK" response to the API consumer and includes in the message content body a representation of the replaced Subscription resource. Alternatively, the PIM can also return a "204 No Content" response to the API consumer indicating successful operation.
14. The API consumer sends a DELETE request to the Individual Subscription resource.
15. The PIM deletes the individual Subscription resource.
16. The PIM returns a "204 No Content" response to the API consumer indicating that the resource has been successfully deleted. Alternatively, the PIM can also return a "200 OK" response to the API consumer including a representation of the deleted Subscription resource in the message content body.

Error handling: The PIM rejects a subscription management request if the subscription representation in the creation or replacement request is not valid, or properties in the modify request are not modifiable, etc. In case of failure, appropriate error information is provided in the response.

A.3 Flow of sending notifications

This clause describes the procedure for sending event notifications to subscribed consumers.

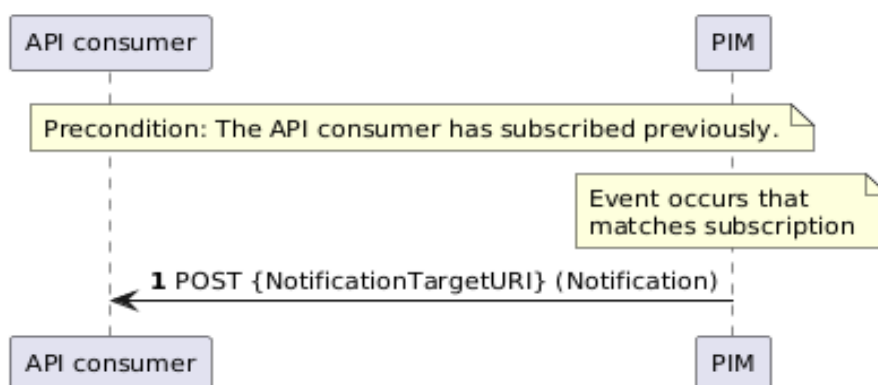


Figure A.3-1: Flow of sending notifications

The procedure consists of the following steps as illustrated in Figure A.3-1.

Precondition: The API consumer has subscribed previously to receiving notifications related to specific events.

1. If an event occurs that matches the eventing criteria which are part of the subscription, the PIM generates a notification that includes information about the event and sends it in the body of a POST request to the notification target URI which the API consumer had registered as part of the subscription request.

Postcondition: None.

Error handling: None.

A.4 Flow of monitoring tasks related to asynchronous operations (polling method)

This clause describes the procedure for the API consumer to monitor tasks related to asynchronous operations that the PIM has been performing upon receiving a valid request for that operation from the API consumer.

The following sequence refers to the "polling method" where the API consumer periodically polls the Task Monitor resource for monitoring the status of an asynchronous operation.

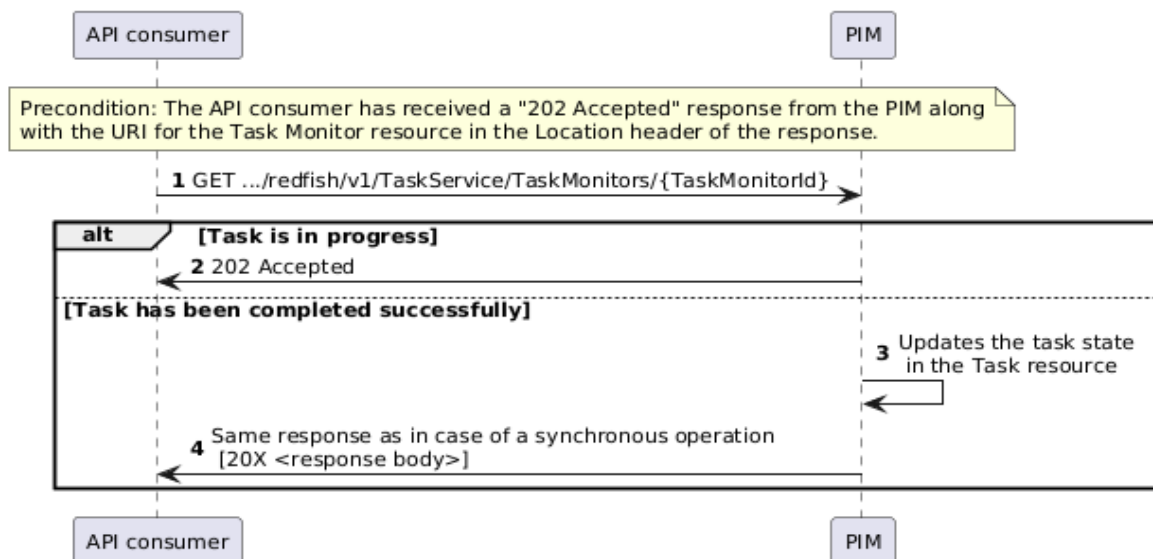


Figure A.4-1: Flow of monitoring asynchronous operation (polling method)

The procedure consists of the following steps as illustrated in Figure A.4-1.

Precondition: The API consumer has received a "202 Accepted" response from the PIM along with the URI for the Task Monitor resource in the Location header of the response.

1. The API consumer sends a GET request to the Individual Task Monitor resource identified by its URI that the consumer has previously received from the API producer in response to the asynchronous operation request.
2. If the PIM is performing the task associated with the requested operation, the API consumer receives the "202 Accepted" response from the PIM.
3. Once the asynchronous operation has been completed successfully, the PIM updates the task state in the corresponding Task resource that is managed by the Task Service.
4. The PIM sends the response to the API consumer with appropriate status code, headers and response body that would have been sent as a synchronous response to the requested operation.

Postcondition: None.

Error handling: In case of failure of the originally requested asynchronous operation, appropriate error information is provided in the response.

A.5 Flow of monitoring tasks related to asynchronous operations (subscribe/notify method)

This clause describes the procedure for the API consumer to monitor tasks related to asynchronous operations that the PIM has been performing upon receiving a valid request for that operation from the API consumer.

The following sequence refers to the "subscribe/notify method" where the API consumer subscribes to state changes in the Task resource associated with the ongoing asynchronous operation. Once the API consumer gets notified about the completion of said Task, it accesses the corresponding Task Monitor resource to get the result of the asynchronous operation.

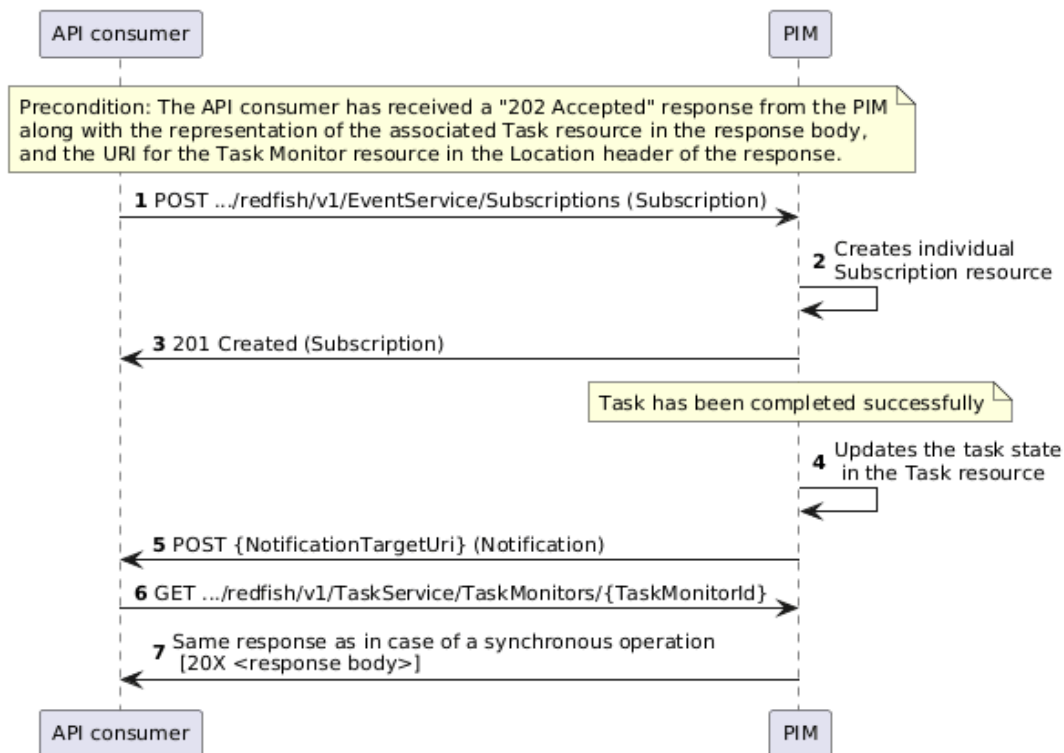


Figure A.5-1: Flow of monitoring asynchronous operation (subscribe/notify method)

The procedure consists of the following steps as illustrated in Figure A.5-1.

Precondition: The API consumer has received a "202 Accepted" response from the PIM along with the representation of the associated Task resource in the response body, and the URI for the Task Monitor resource in the Location header of the response.

1. The API consumer subscribes to get notified about completion of the task associated with asynchronous operation using the Redfish® Event Service.
2. The PIM creates the subscription.
3. The PIM sends the response to API consumer indicating successful subscription creation.
4. Once the asynchronous operation has been completed successfully, the PIM updates the task state in the corresponding Task resource that is managed by the Task Service.
5. The API consumer receives an event notification about completion of the task associated with the asynchronous operation.
6. The API consumer sends a GET request to the Individual Task Monitor resource identified by its URI that the consumer has previously received from the PIM in the "202 Accepted" response to the asynchronous operation request (see precondition).

7. The PIM sends the response to the API consumer with appropriate status code, headers and response body that would have been sent as a synchronous response to the requested operation.

Postcondition: None.

Error handling: In case of failure of the originally requested asynchronous operation, appropriate error information is provided in the response.

Annex B (informative): Relationship between the PIM and the standard Redfish® service

B.1 Introduction

According to the PIM architectural framework and the information model specified in the ETSI GS NFV-IFA 053 [1], the PIM function can manage multiple NFVI-PoPs and the physical resources within those NFVI-PoPs. In such a scenario, the PIM can provide a centralized entry point for managing the physical infrastructure of each NFVI-PoP. The Redfish® standard on the other hand specifies a management interface and data model without mandating a particular architectural development, i.e. it is agnostic to whether management is implemented in a centralized or distributed manner. In practise, a Redfish® service can be seen as responsible for managing the hardware resources within an individual NFVI-PoP, or a site. This relationship is demonstrated with the help of an example manifestation of a distributed architecture comprising of multiple NFVI-PoPs as illustrated in Figure B.1-1.

NOTE: Within the NFV framework, the concept of NFVI-PoP represents a physical site or a Point of Presence (PoP) that offers infrastructure (NFVI) for the NFV workloads. The NFVI can further contain different kinds of infrastructure resources, e.g. physical, virtual, CIS cluster. See ETSI GR NFV 003 [i.1] for more clarification on the concepts of NFVI, NFVI-PoP and examples of various kinds of NFVI resources.

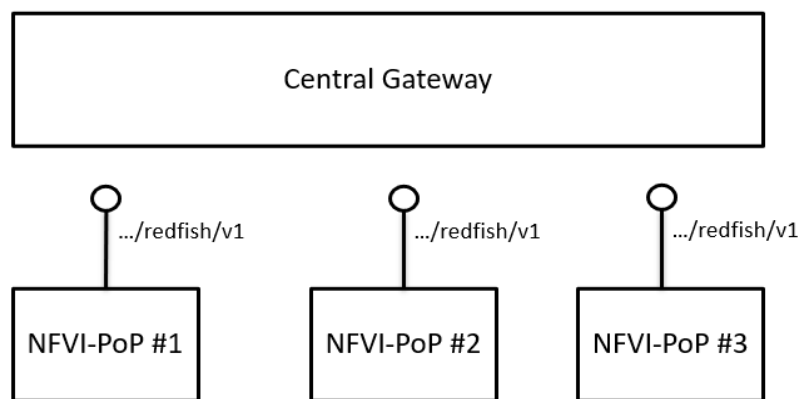


Figure B.1-1: Relationship between the PIM and Redfish® services in a distributed architecture

In a distributed scenario, the PIM function acts as a central gateway towards individual Redfish® services running within each NFVI-PoP, which offer entry points to managing the NFVI physical resources via their corresponding service roots ".../redfish/v1". Subsequent Redfish® resources and services (such as Chassis, Systems, Managers, CompositionService, TelemetryService, EventService, LogService) for a corresponding NFVI-PoP can be reached under the root URI preceded by the appropriate address space of the respective NFVI-PoP, e.g. the IP address and port on which the Redfish® service is exposed on.

An example of an implementation leveraging the Redfish® interface in a distributed architecture is provided by the Open Distributed Infrastructure Management (ODIM) project [i.3]. ODIM provides an example of a resource aggregation function that offers a centralized management interface while interacting with distributed infrastructure resources via Redfish® APIs.

B.2 Management considerations for the distributed architecture

B.2.1 General considerations

The different PIM service interfaces specified in the present document are specified from the perspective of an individual NFVI-PoP. The physical resources within an NFVI-PoP are managed via the Redfish® protocol and its applicable API endpoints that are profiled to realize different PIM service interfaces. The root URI (".../redfish/v1") for these API endpoints stays the same for an NFVI-PoP. In case of multiple NFVI-PoPs, the PIM identifies the appropriate Redfish® service and its root URI for managing the physical resources across NFVI-PoPs.

B.2.2 Considerations for inventory management

The PhysicalResourceInventory information model specified in ETSI GS NFV-IFA 053 [1] represents inventory related information for all the NFVI-PoPs managed by the PIM. In a distributed scenario concerning multiple NFVI-PoPs where each NFVI-PoP is managed by its own Redfish® service, the overall physical resource inventory information can be represented as a collection of the respective NFVI-PoP data objects (see clause 5.3.4 for the definition of NFVI-PoP object).

Retrieving and updating inventory information across different NFVI-PoPs managed by the PIM requires the PIM to identify the appropriate Redfish® services of corresponding NFVI-PoPs and invoke inventory management operations towards those services on a per NFVI-PoP basis. Furthermore, a central aggregation layer is required at the PIM which offers a consolidated view of physical resource inventory across different NFVI-PoPs. This version of the present document does not specify any normative provisions related to this aggregation aspect.

According to the approach mentioned above, an example definition of the physical resource inventory object corresponding to PhysicalResourceInventory in ETSI GS NFV-IFA 053 [1] is demonstrated in Table B.2.2-1. However, such object is not managed via the physical resource inventory management service interface in this version of the present document.

Table B.2.2-1: Example definition of the Physical resource inventory object

Attribute Name	Data Type	Cardinality	Description
id	Identifier	1	Identifier of the resource.
nfviPops	NFVI-PoP	1..N	List of relevant NFVI-PoP data objects.

Annex C (informative): Change history

Date	Version	Information about changes
November 2024	V0.0.1	Initial skeleton of the GS.
December 2024	V0.0.2	Implements following contributions: NFVSOL(24)000448, NFVSOL(24)000449r1, NFVSOL(24)000456r1
June 2025	V0.0.3	Implements following contributions: NFVSOL(25)000181r3, NFVSOL(25)000218r1, NFVSOL(25)000192r1
August 2025	V0.0.4	Implements following contributions: NFVSOL(25)000260r2, NFVSOL(25)000261r1, NFVSOL(25)000274r1
September 2025	V0.0.5	Implements following contributions: NFVSOL(25)000300, NFVSOL(25)000301r1, NFVSOL(25)000302r1, NFVSOL(25)000311
October 2025	V0.0.6	Implements following contributions: NFVSOL(25)000335, NFVSOL(25)000336, NFVSOL(25)000337, NFVSOL(25)000338
November 2025	V0.0.7	The first version created using the SOL026 repository on ETSI Forge, no technical changes on top of v0.0.6
November 2025	V0.0.8	Implements following contributions: NFVSOL(25)000393, NFVSOL(25)000402, NFVSOL(25)000404
November 2025	V0.0.9	Implements following contributions: NFVSOL(25)000411r1, NFVSOL(25)000412
December 2025	V0.0.10	Implements following contributions: NFVSOL(25)000457, NFVSOL(25)000458r1, NFVSOL(25)000459, NFVSOL(25)000460
January 2026	V0.0.11	Implements following contributions: NFVSOL(26)000025, NFVSOL(26)000034, NFVSOL(26)000035, NFVSOL(26)000036
February 2026	V0.0.12	Implements following contributions: NFVSOL(26)000052, NFVSOL(26)000053, NFVSOL(26)000061, NFVSOL(26)000074, NFVSOL(26)000063, Implements following rapporteur action: removal of placeholder Editor's Notes
March 2026	V0.0.13	Implements following contributions: NFVSOL(26)000088, NFVSOL(26)000086, NFVSOL(26)000083, NFVSOL(26)000098, NFVSOL(26)000089

Date	Version	Information about changes
April 2026	V0.0.14	Implements following contributions: NFVSOL(26)000109, NFVSOL(26)000112, NFVSOL(26)000114, NFVSOL(26)000115r1, Implements following rapporteur action: Correction of wrong visualization for Tables 7.5.2-1, 7.5.3-1 and 7.3-1
April 2026	V0.0.15	Implements feedback received during Final review: correction of typo in Annex B "preceeded" to "preceded"
June 2026	V0.0.16	Editorial comments
July 2026	V6.1.1	First published version

History

Version	Date	Status
V6.1.1	July 2026	Publication