Oracle® Healthcare Master Person Index

Relationship Management REST APIs Reference Guide

Release 4.0

E71322-01

February 2016

Oracle Healthcare Master Person Index Relationship Management REST APIs Reference Guide, Release 4.0

E71322-01

Contents

7 Stats Resources

8 Secure Development

A References

Preface

This document contains information on public Application Programming Interfaces (APIs) that are used to interface with Oracle Healthcare Master Person Index (OHMPI) Relationship Management (RM).

Audience

This document is intended for users who are working on OHMPI RM REST APIs.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information and instructions for implementing and using a master person index application, see the following documents in the Oracle Healthcare Master Person Index documentation set:

- *Oracle Healthcare Master Person Index Analyzing and Cleansing Data User's Guide*
- *Oracle Healthcare Master Person Index Australia Patient Solution User's Guide*
- *Oracle Healthcare Master Person Index Command Line Reports and Database Management User's Guide*
- *Oracle Healthcare Master Person Index Configuration Guide*
- *Oracle Healthcare Master Person Index Configuration Reference*
- *Oracle Healthcare Master Person Index Data Manager User's Guide*
- *Oracle Healthcare Master Person Index Installation Guide*
- *Oracle Healthcare Master Person Index Loading the Initial Data Set User's Guide*
- *Oracle Healthcare Master Person Index Match Engine Reference*
- *Oracle Healthcare Master Person Index Message Processing Reference*

- *Oracle Healthcare Master Person Index Provider Index User's Guide*
- *Oracle Healthcare Master Person Index Real-time Loader User's Guide*
- *Oracle Healthcare Master Person Index Relationship Management Data Manager User's Guide*
- *Oracle Healthcare Master Person Index Relationship Management REST APIs Reference Guide*
- *Oracle Healthcare Master Person Index Relationship Management User's Guide*
- *Oracle Healthcare Master Person Index Release Notes*
- *Oracle Healthcare Master Person Index Security Guide*
- *Oracle Healthcare Master Person Index Standardization Engine Reference*
- *Oracle Healthcare Master Person Index Third Party Licenses and Notices*
- *Oracle Healthcare Master Person Index United Kingdom Patient Solution User's Guide*
- *Oracle Healthcare Master Person Index United States Patient Solution User's Guide*
- *Oracle Healthcare Master Person Index User's Guide*
- *Oracle Healthcare Master Person Index Working With HPD Profile Application User's Guide*
- *Oracle Healthcare Master Person Index Working With IHE Profiles User's Guide*

> **Note:** These documents are designed to be used together when implementing a master index application.

Finding Information and Patches on My Oracle Support

Your source for the latest information about Oracle Healthcare Master Person Index is Oracle Support's self-service Web site My Oracle Support (formerly MetaLink).

Before you install and use Oracle Healthcare Master Person Index, always visit the My Oracle Support Web site for the latest information, including alerts, White Papers, installation verification (smoke) tests, bulletins, and patches.

Creating a My Oracle Support Account

You must register at My Oracle Support to obtain a user name and password account before you can enter the Web site.

To register for My Oracle Support:

1. Open a Web browser to `https://support.oracle.com`.
2. Click the **Register here** link to create a My Oracle Support account. The registration page opens.
3. Follow the instructions on the registration page.

Signing In to My Oracle Support

To sign in to My Oracle Support:

1. Open a Web browser to `https://support.oracle.com`.
2. Click **Sign In.**
3. Enter your user name and password.

4. Click **Go** to open the My Oracle Support home page.

Finding Information on My Oracle Support

There are many ways to find information on My Oracle Support.

Searching by Article ID

The fastest way to search for information, including alerts, White Papers, installation verification (smoke) tests, and bulletins is by the article ID number, if you know it.

To search by article ID:

1. Sign in to My Oracle Support at `https://support.oracle.com`.
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the sources icon to the left of the search box, and then select **Article ID** from the list.
4. Enter the article ID number in the text box.
5. Click the magnifying glass icon to the right of the search box (or press the Enter key) to execute your search.

   The Knowledge page displays the results of your search. If the article is found, click the link to view the abstract, text, attachments, and related products.

Searching by Product and Topic

You can use the following My Oracle Support tools to browse and search the knowledge base:

- Product Focus — On the Knowledge page under Select Product, type part of the product name and the system immediately filters the product list by the letters you have typed. (You do not need to type "Oracle.") Select the product you want from the filtered list and then use other search or browse tools to find the information you need.
- Advanced Search — You can specify one or more search criteria, such as source, exact phrase, and related product, to find information. This option is available from the **Advanced** link on almost all pages.

Finding Patches on My Oracle Support

Be sure to check My Oracle Support for the latest patches, if any, for your product. You can search for patches by patch ID or number, or by product or family.

To locate and download a patch:

1. Sign in to My Oracle Support at `https://support.oracle.com`.
2. Click the **Patches & Updates** tab. The Patches & Updates page opens and displays the Patch Search region. You have the following options:
   - In the **Patch ID or Number is** field, enter the number of the patch you want. (This number is the same as the primary bug number fixed by the patch.) This option is useful if you already know the patch number.
   - To find a patch by product name, release, and platform, click the **Product or Family** link to enter one or more search criteria.
3. Click **Search** to execute your query. The Patch Search Results page opens.
4. Click the patch ID number. The system displays details about the patch. In addition, you can view the Read Me file before downloading the patch.

5. Click **Download.** Follow the instructions on the screen to download, save, and install the patch files.

Finding Oracle Documentation

The Oracle Web site contains links to all Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences Documentation

To get user documentation for Oracle Health Sciences applications, go to the Oracle Health Sciences documentation page at:

`http://www.oracle.com/technetwork/documentation/hsgbu-154445.html`

> **Note:** Always check the Oracle Health Sciences Documentation page to ensure you have the latest updates to the documentation.

Finding Other Oracle Documentation

To get user documentation for other Oracle products:

1. Go to the following Web page:

   `http://www.oracle.com/technology/documentation/index.html`

   Alternatively, you can go to `http://www.oracle.com`, point to the Support tab, and then click **Documentation.**

2. Scroll to the product you need and click the link.

3. Click the link for the documentation you need.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Overview

The OHMPI RM REST APIs are uniform interfaces for identifying and managing generic resources of metadata, entities, and relationships. The OHMPI RM REST APIs are used for accessing the functionality of the relationship management service and integrating with the OHMPI applications and any other integral enterprise applications by combining with SOA. The OHMPI RM REST APIs are JAX-RS 2.0 standard RESTful web services. This guide provides the information of the RESTful web services that are available with the current release.

The managed resources of the OHMPI RM application are categorized into the following:

- **Metadata Resources**: Manages relationship types and entity types
- **Relationship Resources**: Manages relationships
- **Entity Resources**: Manages entities
- **MPI Integration Resources**: Manages MPI integration functionality between the RM and the MPI applications
- **Application Management Resources**: Manages domains, rulesets, and tasks
- **Application Statistics Resources**: Manages the application statistical information

The relationship management REST API context root is:

```
http(s)://<server>:<port>/ohmpi-rm/1.0.0
```

1.0.0 is the current version of the relationship management APIs.

1.1 HTTP Verbs' Usage

- GET is used for retrieving and searching resources
- PUT is used for updating the existing resources
- POST is used for creating new resources. It is also used for querying resources by complicated search criteria with sensitive information.

1.2 Generic RM Resource Naming Conventions

- Non-case sensitive
- Start with alphabets
- Mix with any letters, digits, or underscores

- Avoid one-character names
- No two resources have the same resource name
- Each unique resource has only one root resource URI
- Any names of parameters and variables are non-case sensitive
- Any values of parameters and variables are non-case sensitive

2 Metadata Resources

The metadata resources contain the following APIs:

- GET /metadata
- GET /metadata/entities
- GET /metadata/entities/{name}
- POST /metadata/entities/{name}
- PUT /metadata/entities/{name}
- PUT /metadata/entities/{name}/{attribute}
- GET /metadata/relationships
- GET /metadata/relationships/{name}
- POST /metadata/relationships/{name}
- PUT /metadata/relationships/{name}
- PUT /metadata/relationships/{name}/{attribute}
- POST /metadata/logout

2.1 GET /metadata

This API is used to get metadata REST API versioning information and user roles.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
roles	true or false	Option for including user ID and user roles in the response. The default option is false. If the option is not given, the default option is used.

Response Parameters

Parameter	Description
version	OHMPI-RM version number
description	The description of metadata resources
url	The URL of this REST API
roles	When the request roles' parameter is set to true, the highest role of the logged-in user is displayed.
userId	The logged-in user ID

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata?roles=true

Sample Response Body

```
{
 "version": "1.0.0",
 "description": "OHMPI RM Entity and Relationship Metadata Resources",
 "url": "http://localhost:7001/ohmpi-rm/1.0.0/metadata?roles=true",
 "roles": "RmAdministrator",
 "userId": "rm1"
}
```

Sample Response Code

200

2.2 GET /metadata/entities

This API is used to retrieve the list of entity types.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/entities

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
status	all, inactive, or active	Filter entity types based on the status.

Parameter	Value	Description
expand	all	View more details of the entity type. If expand is not passed, API sends compacted information in response.
totalResults	true or false	View total records count.

Response Parameters

Parameter	Description
items	An array containing entity types.
entity type	For information on the entity type attributes in response, see Section A.2.1, "Entity Type Response Structure".
totalResults	Value is set to the total number of entity types based on status filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities?status=Active&expand=all&totalResults=true

Sample Response Body

```
{
  "items": [
  {
    "attributes": [
    {
      "attributeTypeId": 6,
      "defaultValue": null,
      "description": "First Name",
      "indexed": "Y",
      "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "mandatory": "N",
    "maxValue": null,
    "minValue": null,
    "name": "FirstName",
    "readOnly": "N",
    "sizeValue": 15,
    "status": "Active",
  "type": "String"
},
{
  "attributeTypeId": 7,
  "defaultValue": null,
  "description": "Last Name",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
    "mediaType": "application/json",
    "rel": "canonical"
  }
  ],
  "mandatory": "N",
  "maxValue": null,
  "minValue": null,
  "name": "LastName",
  "readOnly": "N",
  "sizeValue": 20,
  "status": "Active",
  "type": "String"
},
{
  "attributeTypeId": 8,
  "defaultValue": null,
  "description": "Gender",
  "indexed": "Y",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "mandatory": "N",
  "maxValue": null,
  minValue": null,
  "name": "Gender",
  "readOnly": "N",
  "sizeValue": 10,
  "status": "Active",
  "type": "String"
},
{
  "attributeTypeId": 9,
  "defaultValue": null,
  "description": "Specialty",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"mandatory": "N",
"maxValue": null,
"minValue": null,
"name": "Specialty",
  "readOnly": "N",
  "sizeValue": 25,
  "status": "Active",
  "type": "String"
},
{
  "attributeTypeId": 10,
  "defaultValue": null,
  "description": "Specialty License",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
    "mediaType": "application/json",
    "rel": "canonical"
  }
 ],
 "mandatory": "N",
 "maxValue": null,
 "minValue": null,
 "name": "License",
 "readOnly": "N",
 "sizeValue": 15,
 "status": "Active",
 "type": "String"
}
],
"description": "IndProvider Entity Type",
"entityTypeId": 2,
"id": "EUID",
"links": [
{
  "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
  "mediaType": "application/json",
  "rel": "canonical"
 }
 ],
 "name": "IndProvider",
 "status": "Active"
},
{
  "attributes": [
    {
      "attributeTypeId": 1,
      "defaultValue": null,
      "description": "First Name",
      "indexed": "Y",
      "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "mandatory": "N",
    "maxValue": null,
    "minValue": null,
    "name": "FirstName",
    "readOnly": "N",
    "sizeValue": 15,
    "status": "Active",
    "type": "String"
  },
{
  "attributeTypeId": 2,
  "defaultValue": null,
  "description": "Last Name",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
    "mediaType": "application/json",
    rel": "canonical"
  }
 ],
 "mandatory": "N",
 "maxValue": null,
 "minValue": null,
 "name": "LastName",
 "readOnly": "N",
 "sizeValue": 20,
 "status": "Active",
 "type": "String"
},
{
  "attributeTypeId": 3,
  "defaultValue": null,
  "description": "Gender",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
    "mediaType": "application/json",
    "rel": "canonical"
  }
 ],
 "mandatory": "N",
 "maxValue": null,
 "minValue": null,
 "name": "Gender",
 "readOnly": "N",
 "sizeValue": 10,
 "status": "Active",
 "type": "String"
},
{
  "attributeTypeId": 4,
  "defaultValue": null,
  "description": "City of Residence",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
    "mediaType": "application/json",
    "rel": "canonical"
  }
 ],
 "mandatory": "N",
 "maxValue": null,
 "minValue": null,
 "name": "City",
 "readOnly": "N",
 "sizeValue": 20,
 "status": "Active",
 "type": "String"
},
{
  "attributeTypeId": 5,
  "defaultValue": null,
  "description": "License of PCP",
  "indexed": "Y",
  "links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
    "mediaType": "application/json",
    "rel": "canonical"
   }
 ],
 "mandatory": "N",
 "maxValue": null,
 "minValue": null,
 "name": "Doctor",
 "readOnly": "N",
 "sizeValue": 15,
 "status": "Active",
 "type": "String"
}
],
"description": "USPatient Entity Type",
"entityTypeId": 1,
"id": "EUID",
"links": [
{
    "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
    "mediaType": "application/json",
    "rel": "canonical"
   }
  ],
 "name": "USPatient",
 "status": "Active"
 }
],
"links": [
  {
    "href": "/ohmpi-rm/1.0.0/metadata/entities",
    "mediaType": "application/json",
    "rel": "canonical"
   }
  ],
  "totalResults": 2
}
```

Sample Response Code

200

2.3 GET /metadata/entities/{name}

This API is used to retrieve the details of an existing entity type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/entities/{name}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Name of the entity type to be searched.
expand	all	View more details of the entity type. If expand is not passed, API sends compacted information in response.

Response Parameters

Parameter	Description
entity type	For information on the entity type attributes in response, see Section A.2.1, "Entity Type Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/IndProvider?expand=all

Sample Response Body

```
{
  "attributes": [
    {
      "attributeTypeId": 6,
      "defaultValue": null,
      "description": "First Name",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "FirstName",
      "readOnly": "N",
      "sizeValue": 15,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 7,
      "defaultValue": null,
      "description": "Last Name",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "LastName",
      "readOnly": "N",
      "sizeValue": 20,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 8,
      "defaultValue": null,
      "description": "Gender",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "Gender",
      "readOnly": "N",
      "sizeValue": 10,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 9,
      "defaultValue": null,
      "description": "Specialty",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "Specialty",
      "readOnly": "N",
      "sizeValue": 25,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 10,
      "defaultValue": null,
      "description": "Specialty License",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "License",
      "readOnly": "N",
      "sizeValue": 15,
      "status": "Active",
      "type": "String"
    }
  ],
  "description": "IndProvider Entity Type",
  "entityTypeId": 2,
  "id": "EUID",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "status": "Active"
}
```

Sample Response Code

200

2.4 POST /metadata/entities/{name}

This API is used to create a new entity type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/entities/{name}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Name of the entity type to be created.
entityTypeResource	-	Details of the entity type to be created. For details on the fields for the entity type request body, see Section A.1.1, "Entity Type Resource".

Response Parameters

Parameter	Description
entity type	For information on the entity type attributes in response, see Section A.2.1, "Entity Type Response Structure". To display all the attributes of the entity type, use the GET /metadata/entities/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/PatientNew

Sample Request Body

```
{
"name": "PatientNew",
"description": "PatientNew Entity Type",
"id": "EUID",
"attributes": [
  {
    "name": "FirstName",
    "type": "String",
    "sizeValue": "80",
    "defaultValue": "Unknown"
  },
  {
    "name": "Age",
    "description": "Patient Age",
    "type": "Integer",
    "minValue" : "0",
    "maxValue" : "200"
  }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "description": null,
  "entityTypeId": 41,
  "id": null,
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "PatientNew",
  "status": "Active"
}
```

Sample Response Code

200

2.5 PUT /metadata/entities/{name}

This API is used to update an existing entity type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/entities/{name}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Name of the entity type to be updated.
status	active or inactive	To activate or deactivate the entity type. When status is passed, the body (entityTypeResource) details are ignored. But, the body must be passed as {}.

Parameter	Value	Description
entityTypeResource	-	Details of the entity type to be updated. This is considered for updating the entity type if the status parameter is not set to any value. For details on the fields for the entity type request body, see Section A.1.1, "Entity Type Resource".

Response Parameters

Parameter	Description
entity type	For information on the entity type attributes in response, see Section A.2.1, "Entity Type Response Structure". To display all the attributes of the entity type, use the GET /metadata/entities/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

2.5.1 To Update Entity Type Status

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/PatientNew?status=Active

Sample Request Body

{ }

Sample Response Body

```
{
  "attributes": [
    {
      "attributeTypeId": 61,
      "defaultValue": "Unknown",
      "description": null,
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "FirstName",
      "readOnly": "N",
      "sizeValue": 80,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 62,
      "defaultValue": null,
      "description": "Patient Age",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": "200",
      "minValue": "0",
      "name": "Age",
      "readOnly": "N",
      "sizeValue": 3,
      "status": "Active",
      "type": "Integer"
    }
  ],
  "description": null,
  "entityTypeId": 41,
  "id": "EUID",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "PatientNew",
  "status": "Active"
}
```

Sample Response Code

200

2.5.2 To Update Entity Type Attributes and Other Details

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/PatientNew

Sample Request Body

```
{
"name": "PatientNew",
"description": "PatientNew Entity Type",
"attributes": [
  {
    "name": "FirstName",
    "description": "Patient First Name",
    "sizeValue": "60"
  },
  {
    "name": "Age",
    "maxValue" : "150"
  }
 ]
}
```

Sample Response Body

```
{
  "attributes": [
    {
      "attributeTypeId": 61,
      "defaultValue": "Unknown",
      "description": "Patient First Name",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "First Name",
      "readOnly": "N",
      "sizeValue": 60,
      "status": "Active",
      "type": "String"
    },
    {
      "attributeTypeId": 62,
      "defaultValue": null,
      "description": "Patient Age",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": "150",
      "minValue": "0",
      "name": "Age",
      "readOnly": "N",
      "sizeValue": 3,
      "status": "Active",
      "type": "Integer"
    }
  ],
  "description": "PatientNew Entity Type",
  "entityTypeId": 41,
  "id": "EUID",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "PatientNew",
  "status": "Active"
}
```

Sample Response Code

200

2.6 PUT /metadata/entities/{name}/{attribute}

This API is used to activate or deactivate an entity type attribute.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/entities/{name}/{attribute}

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity Type Name of entity type whose attribute is to be activated or deactivated
attribute	Matching the regex [a-zA-Z][a-zA-Z_0-9.]+	Attribute Name of the entity type attribute which needs to be activated or deactivated
status	active or inactive	To activate or deactivate the entity type attribute

Response Parameters

Parameter	Description
entity type	For information on the entity type attributes in response, see Section A.2.1, "Entity Type Response Structure". To display all the attributes of the entity type, use the GET /metadata/entities/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/PatientNew/Age?status=Inactive

Sample Response Body

```
{
  "attributes": null,
  "description": null,
  "entityTypeId": 41,
  "id": null,
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/PatientNew",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "PatientNew",
  "status": "Active"
}
```

Sample Response Code

200

2.7 GET /metadata/relationships

This API is used to retrieve the list of relationship types.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/relationships

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
status	all, inactive, or active	Filter relationship types based on status
expand	all	View more details of the relationship type. If expand is not passed, API sends compacted information in response.
totalResults	true or false	View total records count

Response Parameters

Parameter	Description
items	An array containing relationship types

Parameter	Description
relationship type	For information on the relationship type attributes in response, see Section A.2.2, "Relationship Type Response Structure".
totalResults	Value is set to the total number of relationship types based on the status filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/relationships?status=Active&expand=all&totalResults=true

Sample Response Body

```
{
  "items": [
    {
      "attributes": null,
      "createdBy": "rm",
      "createdDate": "12/22/2015",
      "description": "Attending Physician Of",
      "direction": "UNIDIRECTIONAL",
      "effectiveDate": "07/15/2015",
      "end": {
        "attributes": null,
        "description": null,
        "entityTypeId": null,
        "id": null,
        "links": [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "name": "USPatient",
        "status": "Active"
      },
      "expirationDate": "01/01/2020",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/attending-physician-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "modifiedBy": "rm",
      "modifiedDate": "12/22/2015",
      "multiplicity": "ONETOMANY",
      "name": "attending-physician-of",
      "relationshipTypeId": 6,
      "start": {
        "attributes": null,
        "description": null,
        "entityTypeId": null,
        "id": null,
        "links": [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "name": "IndProvider",
        "status": "Active"
      },
      "status": "Active",
      "task": null
    }
  ],
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

2.8 GET /metadata/relationships/{name}

This API is used to retrieve the details of an existing relationship type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/relationships/{name}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship Type Name of the relationship type to be searched. **Note**: A "-" (hyphen) is allowed in the relationship type name.

Parameter	Value	Description
expand	all	View more details of the relationship type. If expand is not passed, API sends compacted information in response.

Response Parameters

Parameter	Description
relationship type	For information on the relationship type attributes in response, see Section A.2.2, "Relationship Type Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/entities/IndProvider?expand=all

Sample Response Body

```
{
  "attributes": [
    {
      "attributeTypeId": 14,
      "defaultValue": null,
      "description": "Date of Marriage",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "DateOfMarriage",
      "readOnly": "N",
      "sizeValue": 10,
      "status": "Active",
      "type": "Date"
    }
  ],
  "createdBy": "rm",
  "createdDate": "12/22/2015",
  "description": "Spouse Of",
  "direction": "BIDIRECTIONAL",
  "effectiveDate": "07/15/2015",
  "end": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "USPatient",
    "status": "Active"
  },
  "expirationDate": "01/01/2020",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "rm",
  "modifiedDate": "12/22/2015",
  "multiplicity": "ONETOONE",
  "name": "spouse-of",
  "relationshipTypeId": 7,
  "start": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "USPatient",
    "status": "Active"
  },
  "status": "Active",
  "task": null
}
```

Sample Response Code

200

2.9 POST /metadata/relationships/{name}

This API is used to create a new relationship type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/relationships/{name}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship Type Name of the relationship type to be created. **Note**: A "-" (hyphen) is allowed in the relationship type name.
relationshipTypeResource	-	Details of the relationship type to be created. For details on the fields for the relationship type request body, see Section A.1.2, "Relationship Type Resource".

Response Parameters

Parameter	Description
relationship type	For information on the relationship type attributes in response, see Section A.2.2, "Relationship Type Response Structure". To display all the attributes of the relationship type, use the GET /metadata/relationships/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/relationships/patient-of?source=USPatient&target=IndProvider

Sample Request Body

```
{
  "name": "patient-of",
  "start": { "name": "USPatient" },
  "end": { "name": "IndProvider" },
  "description": "Patient-to-Individual Provider relationship type",
  "direction": "UNIDIRECTIONAL",
  "multiplicity": "ONETOMANY",
  "status": "Active",
  "effectiveDate": "01/01/2016",
  "expirationDate": "12/31/2035",
  "attributes": [
    {
      "name": "Location",
      "description": "Provider Location",
      "type": "String",
      "status": "Active",
      "sizeValue": "40",
      "mandatory": "N",
      "readOnly": "N",
      "indexed": "N"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": null,
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "multiplicity": null,
  "name": "patient-of",
  "relationshipTypeId": 21,
  "start": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
      ],
      "name": "USPatient",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  }
```

Sample Response Code

200

2.10 PUT /metadata/relationships/{name}

This API is used to update an existing relationship type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/relationships/{name}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship Type Name of the relationship type to be updated. **Note**: A "-" (hyphen) is allowed in the relationship type name.
status	active or inactive	To activate or deactivate the relationship type. When status is passed, the body (relationshipTypeResource) details are ignored. But, the body must be passed as {}.
relationshipTypeResource	-	Details of the relationship type to be updated. This is considered for updating the relationship type if the status parameter is not set to any value. For details on the fields for the relationship type request body, see Section A.1.2, "Relationship Type Resource".

Response Parameters

Parameter	Description
relationship type	For information on the relationship type attributes in response, see Section A.2.2, "Relationship Type Response Structure". To display all the attributes of the relationship type, use the GET /metadata/relationships/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

2.10.1 To Update Relationship Type Status

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/relationships/patient-of?status=Inactive

Sample Request Body

{ }

Sample Response Body

```
{
  "attributes": [
    {
      "attributeTypeId": 81,
      "defaultValue": null,
      "description": "Provider Location",
      "indexed": "N",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "Location",
      "readOnly": "N",
      "sizeValue": 40,
      "status": "Active",
      "type": "String"
    }
  ],
  "createdBy": "<username>",
  "createdDate": "12/31/2015",
  "description": "Patient-to-Individual Provider relationship type",
  "direction": "UNIDIRECTIONAL",
  "effectiveDate": "01/01/2016",
  "end": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": "12/31/2035",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "<username>",
  "modifiedDate": "12/31/2015",
  "multiplicity": "ONETOMANY",
  "name": "patient-of",
  "relationshipTypeId": 21,
  "start": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "USPatient",
    "status": "Active"
  },
  "status": "Inactive",
  "task": null
}
```

Sample Response Code

200

2.10.2 To Update Relationship Type Attributes and Other Details

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/relationships/patient-of

Sample Request Body

```
{
  "name": "patient-of2",
  "direction": "BIDIRECTIONAL",
  "multiplicity": "MANYTOMANY",
  "effectiveDate": "01/04/2016",
  "expirationDate": "11/21/2040",
  "attributes": [
    {
      "name": "Location",
      "sizeValue": "80",
      "mandatory": "Y",
      "indexed": "Y"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": [
    {
      attributeTypeId": 81,
      "defaultValue": null,
      "description": "Provider Location",
      "indexed": "Y",
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "Y",
      "maxValue": null,
      "minValue": null,
      "name": "Location",
      "readOnly": "N",
      "sizeValue": 80,
      "status": "Active",
      "type": "String"
    }
  ],
  "createdBy": "<username>",
  "createdDate": "12/31/2015",
  "description": "Patient-to-Individual Provider relationship type",
  "direction": "BIDIRECTIONAL",
  "effectiveDate": "01/04/2016",
  "end": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": "11/21/2040",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "<username>",
  "modifiedDate": "12/31/2015",
  "multiplicity": "MANYTOMANY",
  "name": "patient-of",
  "relationshipTypeId": 21,
  "start": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "USPatient",
    "status": "Active"
  },
  "status": "Active",
  "task": null
}
```

Sample Response Code

200

2.11 PUT /metadata/relationships/{name}/{attribute}

This API is used to activate or deactivate a relationship type attribute.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/relationships/{name}/{attribute}

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship Type Name of relationship type whose attribute is to be activated or deactivated. **Note**: A "-" (hyphen) is allowed in the relationship type name.

Parameter	Value	Description
attribute	Matching the regex [a-zA-Z][a-zA-Z_0-9.]+	Attribute Name of the relationship type attribute which needs to be activated or deactivated.
status	active or inactive	To activate or deactivate the relationship type attribute.

Response Parameters

Parameter	Description
relationship type	For information on the relationship type attributes in response, see Section A.2.2, "Relationship Type Response Structure". To display all the attributes of the relationship type, use the GET /metadata/relationships/{name} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/
ohmpi-rm/1.0.0/metadata/relationships/patient-of/Location?status=Inactive

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end": {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": null,
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of2",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "patient-of2",
    "relationshipTypeId": 21,
    "start": {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/USPatient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "USPatient",
      "status": "Active"
    },
    "status": "Active",
    "task": null
}
```

Sample Response Code

200

2.12 POST /metadata/logout

This API is used to logout the currently logged-in user from REST API.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/metadata/logout

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
message	Provides username of the user who was logged out

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Status Code	Description
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/metadata/logout

Sample Response Body

```
{
  "message": "User 'rmAdmin' logged out successfully"
}
```

Sample Response Code

200

3 Management Resources

The management resources contain the following APIs:

- GET /management
- GET /management/domains
- GET /management/domains/{domain}
- POST /management/domains/{domain}
- PUT /management/domains/{domain}
- PUT /management/domains/{domain}/files
- GET /management/rulesets
- GET /management/rulesets/{ruleset}
- POST /management/rulesets/{ruleset}
- PUT /management/rulesets/{ruleset}
- PUT /management/rulesets/{ruleset}/files
- GET /management/tasks
- GET /management/tasks/{task}
- POST /management/tasks/{task}
- PUT /management/tasks/{task}
- PUT /management/tasks/{task}/files

3.1 GET /management

This API is used to get management REST API versioning information.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
version	The OHMPI RM version number
description	The description of management resources
url	The URL of this REST API

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management

Sample Response Body

```
{
  "version": "1.0.0",
  "description": "OHMPI RM Management Resources",
  "url": "http://localhost:7001/ohmpi-rm/1.0.0/management"
}
```

Sample Response Code

200

3.2 GET /management/domains

This API is used to retrieve the list of domains.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/domains

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
expand	all, objectxml, or fields	View more details of the domain. The *objectxml* value displays the complete objectxml file, the *fields* value displays the fields present in the objectxml file, and the *all* value displays the complete objectxml file and the fields.
totalResults	true or false	View total records count

Response Parameters

Parameter	Description
items	An array containing domains
domain	For information on the domain attributes in response, see Section A.2.3, "Domain Response Structure".
totalResults	Value is set to the total number of domains based on the status filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/domains?totalResults=true

Sample Response Body

```
{
  "items":
  [
    {
      "domainId": 2,
      "fields": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=fields}",
      "jndiname": "ejb/IndProviderMasterController",
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/management/domains/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "objectXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=objectxml}",
      "url": "t3://localhost:7001",
      "usercode": "********",
      "userid": "mpiAdmin"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/management/domains",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

3.3 GET /management/domains/{domain}

This API is used to retrieve the details of an existing domain.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/domains/{domain}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
domain	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Domain Name of the domain to be searched
expand	all, objectxml, or fields	View more details of the domain. The *objectxml* value displays the complete objectxml file, the *fields* value displays the fields present in the objectxml file, and the *all* value displays the complete objectxml file and the fields.

Response Parameters

Parameter	Description
domain	For information on the domain attributes in response, see Section A.2.3, "Domain Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/domains/IndProvider

Sample Response Body

```
{
  "domainId": 2,
  "fields": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=fields}",
  "jndiname": "ejb/IndProviderMasterController",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/management/domains/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "objectXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=objectxml}",
  "url": "t3://localhost:7001",
  "usercode": "********",
  "userid": "mpiAdmin"
}
```

Sample Response Code

200

3.4 POST /management/domains/{domain}

This API is used to register a new domain.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/domains/{domain}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
domain	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Domain Name of the domain to be registered
domainResource	-	Details of the domain to be registered. For details on the fields for the domain request body, see Section A.1.3, "Domain Resource".

Response Parameters

Parameter	Description
domain	For information on the domain attributes in response, see Section A.2.3, "Domain Response Structure". To display all the attributes of the domain, use the GET /management/domains/{domain} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/domains/IndProvider

Sample Request Body

```
{
  "name": "IndProvider",
  "jndiname": "ejb/IndProviderMasterController ",
  "usercode": "d2VsY29tZTE=",
  "url": "t3://localhost:7001",
  "userid": "mpiAdmin"
}
```

Sample Response Body

```
{
  "domainId": 2,
  "fields": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=fields}",
  "jndiname": "ejb/IndProviderMasterController",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/management/domains/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "objectXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=objectxml}",
  "url": "t3://localhost:7001",
  "usercode": "********",
  "userid": "mpiAdmin"
}
```

Sample Response Code

200

3.5 PUT /management/domains/{domain}

This API is used to update an existing domain.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/domains/{domain}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
domain	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Domain Name of the domain to be updated
domainResource	-	Details of the domain to be updated. For details on the fields for the domain request body, see Section A.1.3, "Domain Resource".

Response Parameters

Parameter	Description
domain	For information on the domain attributes in response, see Section A.2.3, "Domain Response Structure". To display all the attributes of the domain, use the GET /management/domains/{domain} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/domains/IndProvider

Sample Request Body

```
{
  "name": "IndProvider",
  "jndiname": "ejb/IndProviderMasterController ",
  "usercode": "d2VsY29tZTE=",
  "url": "t3://localhost:7011",
  "userid": "mpiAdminNew"
}
```

Sample Response Body

```
{
  "domainId": 2,
  "fields": "{canonicalLink:
```

```
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=fields}",
  "jndiname": "ejb/IndProviderMasterController",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/management/domains/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "objectXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=objectxml}",
  "url": "t3://localhost:7011",
  "usercode": "********",
  "userid": "mpiAdminNew"
}
```

Sample Response Code

200

3.6 PUT /management/domains/{domain}/files

This API is used to upload object definition xml file for a domain.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/domains/{domain}/files

Request Content Type

multipart/form-data

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Parameters

Parameter	Description
domain	Domain Name of the domain whose object.xml file needs to be uploaded
objectFile	The XML file to be uploaded
contentDisposition	MIME content-disposition header which contains name of the uploaded file.

Response Parameters

Parameter	Description
domain	For information on the domain attributes in response, see Section A.2.3, "Domain Response Structure". To display all the attributes of the domain, use the GET /management/domains/{domain} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/domains/IndProvider/files

Sample Response Body

```
{
  "domainId": 2,
  "fields": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=fields}",
  "jndiname": "ejb/IndProviderMasterController",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/management/domains/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "objectXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/domains/IndProvider?expand=objectxml}",
  "url": "t3://localhost:7011",
  "usercode": "********",
  "userid": "mpiAdminNew"
}
```

Sample Response Code

200

3.7 GET /management/rulesets

This API is used to retrieve the list of rulesets.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/rulesets

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
expand	all or rulexml	View more details of the ruleset. The *all* or *rulexml* value displays the complete xml file of the ruleset.
totalResults	true or false	View total records count

Response Parameters

Parameter	Description
items	An array containing rulesets
ruleset	For information on the ruleset attributes in response, see Section A.2.4, "Ruleset Response Structure".
totalResults	Value is set to the total number of rulesets based on the status filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/rulesets?totalResults=true

Sample Response Body

```
{
  "items": [
  {
    "createdBy": "rmAdmin",
    "createdDate": "01/08/2016",
    "description": "Relationship ruleset between Patient and IndProvider",
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": "rmAdmin",
    "modifiedDate": "01/08/2016",
    "name": "CreateRelationshipRuleset",
    "rulesetId": 1,
    "ruleXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset?expand=ruleXml}",
    "source": "Patient",
    "status": "Active",
    "target": "IndProvider"
```

```
      }
    ],
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/management/rulesets",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
    "totalResults": 1
}
```

Sample Response Code

200

3.8 GET /management/rulesets/{ruleset}

This API is used to retrieve the details of an existing ruleset.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/rulesets/{ruleset}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
ruleset	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Ruleset Name of the ruleset to be searched
expand	all or rulexml	View more details of the ruleset. The *all* or *rulexml* value displays the complete xml file of the ruleset.

Response Parameters

Parameter	Description
domain	For information on the ruleset attributes in response, see Section A.2.4, "Ruleset Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset

Sample Response Body

```
{
  "createdBy": "rmAdmin",
  "createdDate": "01/08/2016",
  "description": "Relationship ruleset between Patient and IndProvider",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "rmAdmin",
  "modifiedDate": "01/08/2016",
  "name": "CreateRelationshipRuleset",
  "rulesetId": 1,
  "ruleXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset?expand=ruleXml}",
  "source": "Patient",
  "status": "Active",
  "target": "IndProvider"
}
```

Sample Response Code

200

3.9 POST /management/rulesets/{ruleset}

This API is used to register a new ruleset.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/rulesets/{ruleset}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
ruleset	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Ruleset Name of the ruleset to be registered

Parameter	Value	Description
rmRuleSetResource	-	Details of the ruleset to be registered. For details on the fields for the ruleset request body, see Section A.1.4, "RmRuleset Resource".

Response Parameters

Parameter	Description
ruleset	For information on the ruleset attributes in response, see Section A.2.4, "Ruleset Response Structure". To display all the attributes of the domain, use the GET /management/rulesets/{ruleset} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset

Sample Request Body

```
{
  "name": "CreateRelationshipRuleset",
  "description": "Relationship ruleset between Patient and IndProvider",
  "source": "Patient",
  "target": "IndProvider",
  "status": "Active"
}
```

Sample Response Body

```
{
  "createdBy": "rmAdmin",
  "createdDate": "01/08/2016",
  "description": "Relationship ruleset between Patient and IndProvider",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "rmAdmin",
  "modifiedDate": "01/08/2016",
  "name": "CreateRelationshipRuleset",
  "rulesetId": 1,
  "ruleXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset?expand=ruleXml}",
    "source": "Patient",
    "status": "Active",
    "target": "IndProvider"
  }
```

Sample Response Code

200

3.10 PUT /management/rulesets/{ruleset}

This API is used to update an existing ruleset.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/rulesets/{ruleset}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
ruleset	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Ruleset Name of the ruleset to be updated
rmRuleSetResource	-	Details of the ruleset to be updated. For details on the fields for the ruleset request body, see Section A.1.4, "RmRuleset Resource".

Response Parameters

Parameter	Description
ruleset	For information on the ruleset attributes in response, see Section A.2.4, "Ruleset Response Structure". To display all the attributes of the ruleset, use the GET /management/rulesets/{ruleset} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset

Sample Request Body

```
{
  "name": "CreateRelationshipRuleset",
  "description": "Relationship ruleset between OrgProvider and IndProvider",
  "source": "OrgProvider"
}
```

Sample Response Body

```
{
  "createdBy": "rmAdmin",
  "createdDate": "01/08/2016",
  "description": "Relationship ruleset between OrgProvider and IndProvider",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "rmAdmin",
  "modifiedDate": "01/08/2016",
  "name": "CreateRelationshipRuleset",
  "rulesetId": 1,
  "ruleXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset?expand=ruleXml}",
  "source": "OrgProvider ",
  "status": "Active",
  "target": "IndProvider"
}
```

Sample Response Code

200

3.11 PUT /management/rulesets/{ruleset}/files

This API is used to upload ruleset xml file.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/rulesets/{ruleset}/files

Request Content Type

multipart/form-data

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Parameters

Parameter	Description
ruleset	Ruleset Name of ruleset whose xml file needs to be uploaded
ruleSetFile	The XML file to be uploaded
contentDisposition	MIME content-disposition header which contains name of the uploaded file.

Response Parameters

Parameter	Description
ruleset	For information on the ruleset attributes in response, see Section A.2.4, "Ruleset Response Structure". To display all the attributes of the ruleset, use the GET /management/rulesets/{ruleset} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset/files

Sample Response Body

```
{
  "createdBy": "rmAdmin",
  "createdDate": "01/08/2016",
  "description": "Relationship ruleset between Patient and IndProvider",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "rmAdmin",
  "modifiedDate": "01/08/2016",
  "name": "CreateRelationshipRuleset",
  "rulesetId": 1,
  "ruleXml": "{canonicalLink:
/ohmpi-rm/1.0.0/management/rulesets/CreateRelationshipRuleset?expand=ruleXml}",
  "source": "Patient",
  "status": "Active",
  "target": "IndProvider"
}
```

Sample Response Code

200

3.12 GET /management/tasks

This API is used to retrieve the list of tasks.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/tasks

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
expand	all or script	View more details of the task. The *all* or *script* value displays the complete groovy script file of the task.
totalResults	true or false	View total records count

Response Parameters

Parameter	Description
items	An array containing tasks
task	For information on the task attributes in response, see Section A.2.5, "Task Response Structure".
totalResults	Value is set to the total number of tasks based on the status filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/tasks?totalResults=true

Sample Response Body

```
{
  "items": [
  {
  {
    "description": "Task to create relationship",
    "links": [
      {
        "href": "/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "CreateRelationshipTask ",
    "script": "{canonicalLink:
/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask?expand=script}",
    "status": "Active",
    "taskId": 1
  }
  ],
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/tasks",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

3.13 GET /management/tasks/{task}

This API is used to retrieve the details of an existing task.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/tasks/{task}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
task	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Task Name of the task to be searched
expand	all or script	View more details of the task. The *all* or *script* value displays the complete groovy script file of the task.

Response Parameters

Parameter	Description
task	For information on the task attributes in response, see Section A.2.5, "Task Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask

Sample Response Body

```
{
  "description": "Task to create relationship",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "CreateRelationshipTask ",
  "script": "{canonicalLink:
/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask?expand=script}",
  "status": "Active",
  "taskId": 1
}
```

Sample Response Code

200

3.14 POST /management/tasks/{task}

This API is used to register a new task.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/tasks/{task}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
task	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Task Name of the task to be registered
rmTaskResource	-	Details of the task to be registered. For details on the fields for the task request body, see Section A.1.5, "RmTask Resource".

Response Parameters

Parameter	Description
task	For information on the task attributes in response, see Section A.2.5, "Task Response Structure". To display all the attributes of the task, use the GET /management/tasks/{task} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/tasks/CreateRelationship

Sample Request Body

```
{
  "name": "CreateRelationshipTask",
  "description": "Task to create relationship",
  "status": "Active"
}
```

Sample Response Body

```
{
  "description": "Task to create relationship",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "CreateRelationshipTask ",
  "script": "{canonicalLink:
/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask?expand=script}",
  "status": "Active",
  "taskId": 1
}
```

Sample Response Code

200

3.15 PUT /management/tasks/{task}

This API is used to update an existing task.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/tasks/{task}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
task	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Task Name of the task to be updated
rmTaskResource	-	Details of the task to be updated. For details on the fields for the task request body, see Section A.1.5, "RmTask Resource".

Response Parameters

Parameter	Description
task	For information on the task attributes in response, see Section A.2.5, "Task Response Structure". To display all the attributes of the task, use the GET /management/tasks/{task} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask

Sample Request Body

```
{
  "name": "CreateRelationshipTask",
  "description": "Task to create relationship in RM Database"
}
```

Sample Response Body

```
{
  "description": "Task to create relationship in RM Database ",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "CreateRelationshipTask ",
  "script": "{canonicalLink:
/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask?expand=script}",
  "status": "Active",
  "taskId": 1
}
```

Sample Response Code

200

3.16 PUT /management/tasks/{task}/files

This API is used to upload script file for a task.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/management/tasks/{task}/files

Request Content Type

multipart/form-data

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Parameters

Parameter	Description
task	Task Name of task whose groovy script file needs to be uploaded
rmTaskFile	Groovy script file to be uploaded
contentDisposition	MIME content-disposition header which contains name of the uploaded file.

Response Parameters

Parameter	Description
task	For information on the task attributes in response, see Section A.2.5, "Task Response Structure". To display all the attributes of the task, use the GET /management/tasks/{task} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Resource Not Exists
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask/files

Sample Response Body

```
{
  "description": "Task to create relationship",
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "CreateRelationshipTask ",
  "script": "{canonicalLink:
/ohmpi-rm/1.0.0/management/tasks/CreateRelationshipTask?expand=script}",
  "status": "Active",
  "taskId": 1
}
```

Sample Response Code

200

4

Entities Resources

The entities resources contain the following APIs:

- GET /resources
- GET /resources/entities/{id}
- POST /resources/entities
- POST /resources/entities/{name}/{attribute}
- POST /resources/entities/{name}
- PUT /resources/entities/{id}

4.1 GET /resources

This API is used to retrieve entity and relationship resources REST API version information.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
version	OHMPI RM version number
description	The description of entity and relationship resources
url	The URL of this REST API

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources

Sample Response Body

```
{
  "items":
  [
    {
      "description": "OHMPI RM Entity Resources",
      "roles": null,
      "url": "http://localhost:7001/ohmpi-rm/1.0.0/resources/entities",
      "userId": null,
      "version": "1.0.0"
    },
    {
      "description": "OHMPI RM Relationship Resources",
      "roles": null,
      "url": "http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships",
      "userId": null,
      "version": "1.0.0"
    }
  ]
}
```

Sample Response Code

200

4.2 GET /resources/entities/{id}

This API is used to get entity by its ID.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/entities/{id}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
id	Matching the regex [a-zA-Z0-9]+	EntityId of the entity to be retrieved

Response Parameters

Parameter	Description
entity	For information on the entity attributes in response, see Section A.2.6, "Entity Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Id
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities /1

Sample Response Body

```
{
  "attributes":
  [
    {
      "entity":
      {
        "attributes": null,
        "entityId": 1,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/entities/1",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "status": null,
        "type": null
      },
      "entityAttributeId": 1,
      "name": "FirstName",
      "value": "Mark"
    }
  ],
  "entityId": 1,
  "id": "0000000003",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": "Active",
  "type":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": 1,
    "id": null,
    "links":
```

```
  [
   {
    "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
    "mediaType": "application/json",
    "rel": "canonical"
   }
  ],
  "name": "Patient",
  "status": "Active"
 }
}
```

Sample Response Code

200

4.3 POST /resources/entities

This API is used to search entities by entity type and entity attributes.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/entities

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entities to be searched. It is a mandatory parameter.
status	active, inactive, or all	Status of entities to filter
limit	Any positive integer	Maximum number of results to return in the REST API response. Default value is 1000.
offset	Any positive integer	Offset from which the results should be provided in the response. Default value is 1.
totalResults	true or false	View total records count
expand	all	View more details of the entities
entityResource	-	Details of the entities to be searched. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
items	An array containing entities

Parameter	Description
entity	For information on the entity attributes in response, see Section A.2.6, "Entity Response Structure".
totalResults	Value is set to the total number of entities based on the filter. This might be greater than the number of results displayed in the REST API response, based on offset and limit values. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities?name=Patient&status=active

Sample Request Body

```
{
  "attributes":
  [
    {
      "name":"FirstName",
      "value":"Mark"
    }
  ]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "entityId": 1,
      "id": "0000000003",
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/entities/1",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "status": "Active",
      "type":
      {
        "attributes": null,
        "description": null,
        "entityTypeId": 1,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "name": "Patient",
        "status": "Active"
      }
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

4.4 POST /resources/entities/{name}/{attribute}

This API is used to search entities by entity type and date attribute range.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/entities/{name}/{attribute}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entities to be searched
attribute	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Attribute name of the entity type to be searched using range search. It must be of type Date.
since	Should be in the format yyyy-MM-dd	Start date from which attribute must be searched. It is a mandatory parameter.

Parameter	Value	Description
until	Should be in the format yyyy-MM-dd	End date till which attribute must be searched. Default is today's date.
status	active, inactive, or all	Status of entities to filter
orderBy	Format: fieldname | sortOrder where fieldname can be id, entityId, status; sortOrder can be asc, desc	Order by built-in attribute name. Default is order by entityId in ascending order.
limit	Any positive integer	Maximum number of results to return in the REST API response. Default value is 1000.
offset	Any positive integer	Offset from which the results should be provided in the response. Default value is 1.
totalResults	true or false	View total records count
expand	all	View more details of the entities
entityResource	-	Details of the entities to be searched. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
items	An array containing entities
entity	For information on the entity attributes in response, see Section A.2.6, "Entity Response Structure".
totalResults	Value is set to the total number of entities based on the filter. This might be greater than the number of results displayed in the REST API response, based on the offset and limit values.The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities/IndPatient/DOB?since=2010-01-01

Sample Request Body

```
{
  "attributes":
  [
    {
      "name":"FirstName",
      "value":"Mark"
    }
  ]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "entityId": 5,
      "id": "IP5",
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/entities/IndPatient/DOB/5",
          "mediaType": "application/json",
          "rel": "canonical"
        },
      ],
      "status": "Active",
      "type":
      {
        "attributes": null,
        "description": null,
        "entityTypeId": 1,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/entities/IndPatient",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "name": "IndPatient",
        "status": "Active"
      }
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/IndPatient/DOB",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": null
}
```

Sample Response Code

200

4.5 POST /resources/entities/{name}

This API is used to create an entity of the specific entity type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/entities/{name}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entity to be created
entityResource	-	Details of the entities to be created. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
entity	For information on the entity attributes in response, see Section A.2.6, "Entity Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities/Patient

Sample Request Body

```
{
  "id": "1000000101",
  "type":
  {
    "name": "Patient"
  },
  "attributes":
  [
    {
      "name": "FirstName",
      "value": "Peter"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes":
  [
    {
      "entity": null,
      "entityAttributeId": 1,
      "name": "FirstName",
      "value": "Peter"
    }
  ],
  "entityId": 1,
  "id": "1000000101",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/Patient",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": "Active",
  "type":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": 1,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  }
}
```

Sample Response Code

200

4.6 PUT /resources/entities/{id}

This API is used to update an entity by its ID.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/entities/{id}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
id	Matching the regex [a-zA-Z0-9]+	EntityId of the entity to be updated
status	active or inactive	To activate or deactivate the entity. When status is passed, the body (entityResource) details are ignored. However, the body must be passed as {}.
entityResource	-	Details of the entity to be updated. This is considered for updating the entity if the status parameter is not set to any value. For details on the fields for the relationship type request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
entity	For information on the entity attributes in response, see Section A.2.6, "Entity Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Id
500	Error while processing

4.6.1 To Update Entity Status

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities/1?status=active

Sample Request Body

```
{}
```

Sample Response Body

```
{
  "attributes":
  [
    {
      "entity": null,
      "entityAttributeId": 1,
      "name": "FirstName",
      "value": "Peter"
    }
  ],
  "entityId": 1,
  "id": "1000000101",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": "Active",
  "type":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": 1,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  }
}
```

Sample Response Code

200

4.6.2 To Update Entity Attributes and Other Details

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/entities/1

Sample Request Body

```
{
  "attributes":
  [
    {
      "name": "FirstName",
      "value": "Harry"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes":
  [
    {
      "entity": null,
      "entityAttributeId": 1,
      "name": "FirstName",
      "value": "Harry"
    }
  ],
  "entityId": 1,
  "id": "1000000101",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": "Active",
  "type":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": 1,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  }
}
```

Sample Response Code

200

5 Relationships Resources

The relationships resources contain the following APIs:

- GET /resources/relationships/{id}
- GET /resources/relationships/entities/{entityId}
- GET /resources/relationships/entities/{entity}
- POST /resources/relationships/{entity}
- POST /resources/relationships/{name}/{attribute}
- POST /resources/relationships/{name}/{source}/{target}
- POST /resources/relationships/entities/{entity}
- POST /resources/relationships/entities/{source}/{target}
- PUT /resources/relationships/{id}
- PUT /resources/relationships/entities/{name}

5.1 GET /resources/relationships/{id}

This API is used to retrieve a relationship by its ID.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/{id}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
Id	Matching the regex [a-zA-Z0-9]+	Relationship id of the relationship to be retrieved
expand	all	View more details of the relationship

Response Parameters

Parameter	Description
relationship	For more information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Id
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships /1?expand=all

Sample Response Body

```
{
  "attributes":
  [
    {
      "name": "Location",
      "relationship":
      {
        "attributes": null,
        "createdBy": null,
        "createdDate": null,
        "effectiveDate": null,
        "expirationDate": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/relationships/1",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "relationshipId": 1,
        "source": null,
        "status": null,
        "target": null,
        "type": null,
        "validity": null
      },
      "relationshipAttributeId": 1,
      "value": "Bengaluru"
    }
  ],
  "createdBy": "rmAdmin",
  "createdDate": "2015-12-17 10:43:58",
  "effectiveDate": "2015-12-30 08:07:14",
"expirationDate": "2025-12-30 08:07:14",
"links":
[
  {
    "href": "/ohmpi-rm/1.0.0/resources/relationships/1",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"modifiedBy": "rmAdmin",
"modifiedDate": "2016-01-11 12:08:51",
"relationshipId": 1,
"source":
{
"attributes":
[
  {
    "entity":
    {
      "attributes": null,
      "entityId": 1,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/entities/1",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "status": null,
      "type": null
    },
    "entityAttributeId": 1,
    "name": "FirstName",
    "value": "Dunna"
  }
],
"entityId": 1,
"id": "566022508Aaa",
"links":
[
  {
    "href": "/ohmpi-rm/1.0.0/resources/relationships/1",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"status": "Active",
"type":
{
  "attributes": null,
  "description": null,
  "entityTypeId": 1,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
      "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  }
},
"status": "Active",
"target":
{
"attributes":
[
  {
    "entity":
  {
    "attributes": null,
    "entityId": 2,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/2",
        "mediaType": "application/json",
        "rel": "canonical"
      }
      ],
      "status": null,
      "type": null
    },
    "entityAttributeId": 9,
    "name": "Names.FirstName",
    "value": "Fedrik"
  }
],
"entityId": 2,
"id": "0000626069",
"links":
[
  {
    "href": "/ohmpi-rm/1.0.0/resources/relationships/1",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"status": "Active",
"type":
{
  "attributes": null,
  "description": null,
  "entityTypeId": 2,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "name": "IndProvider",
  "status": "Active"
  }
},
"type":
{
  "attributes":
  [
    {
      "attributeTypeId": 15,
      "defaultValue": null,
      "description": "Patient visit location",
      "indexed": "Y",
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "Y",
      "maxValue": null,
      "minValue": null,
      "name": "Location",
      "readOnly": "N",
      "sizeValue": 70,
      "status": "Active",
      "type": "String"
    }
  ],
  "createdBy": "Admin",
  "createdDate": "2015-07-01 00:00:00",
  "description": "Patient-to-Individual Provider relationship type",
  "direction": "UNIDIRECTIONAL",
  "effectiveDate": "2015-07-10 00:00:00",
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": "2020-07-10 00:00:00",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "Admin",
  "modifiedDate": "2015-07-10 00:00:00",
  "multiplicity": "ONETOMANY",
  "name": "patient-of",
  "relationshipTypeId": 7,
  "start":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  },
  "status": "Active",
  "task": null
 },
 "validity": "Resolved"
}
```

Sample Response Code

200

5.2 GET /resources/relationships/entities/{entityId}

This API is used to search relationships based on entityId.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/entities/{entityId}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entityId	Matching the regex [a-zA-Z0-9]+	EntityId of the entity whose relationships need to be retrieved. If the relationship direction is BIDIRECTIONAL, relationships matching the entity ID from source and target will be retrieved. For UNIDIRECTIONAL, relationships matching the entity ID only from the source will be retrieved.

Parameter	Value	Description
status	active or inactive	Status of the relationships to filter
depth	-	Maximum length of the relationship path search. Currently, not in use.
totalResults	true or false	View total records count
expand	all	View more details of the relationship

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".
totalResults	Value is set to the total number of relationships based on the filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Id
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships /entities/1?totalResults=true

Sample Response Body

```
{
  "items": [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links": [
        {
          "href": "/ohmpi-rm/1.0.0/resources/relationships/4",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "modifiedBy": null,
      "modifiedDate": null,
      "relationshipId": 4,
      "source":
      {
        "attributes": null,
  "entityId": 1,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "multiplicity": null,
        "name": "patient-of",
        "relationshipTypeId": 5,
        "start":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

5.3 GET /resources/relationships/entities/{entity}

This API is used to search relationships based on source or target entity type, entity ID, and relationship type.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/entities/{entity}

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entity	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entity whose relationships need to be retrieved
id	Matching the regex [a-zA-Z0-9]+	ID of the entity. It is mandatory and must be the EUID for the entity type.
label	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name whose relationships need to be retrieved
status	active or inactive	Status of the relationships to filter
depth	-	Maximum length of the relationship path search. Currently, not in use.
totalResults	true or false	View total records count
expand	all	View more details of the relationship

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".
totalResults	Value is set to the total number of relationships based on filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships /entities/Patient

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
```

```
[
  {
    "href": "/ohmpi-rm/1.0.0/resources/relationships/4",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"modifiedBy": null,
"modifiedDate": null,
"relationshipId": 4,
"source":
{
  "attributes": null,
  "entityId": 1,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "IndProvider",
          "status": "Active"
        },
        "expirationDate": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "multiplicity": null,
        "name": "patient-of",
        "relationshipTypeId": 5,
        "start":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/entities/Patient",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": 1
}
```

Sample Response Code

200

5.4 POST /resources/relationships/{entity}

This API is used to search relationships from the source or the target.

Difference between the POST /resources/relationships/{entity} API and the GET /resources/relationships/entities/{entity} API is that you can search by multiple entity attributes by passing them in the body of the REST API. Search in the body is done as:

- If entityId is passed in the body, it will be used for search. All other content will be ignored.
- If entityId is not passed but id is passed, it will be used for search. All other content will be ignored.
- If entityId or id is not passed, then the search will be done based on the attributes passed in the request body.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/{entity}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entity	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entity whose relationships need to be retrieved
label	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name whose relationships need to be retrieved
status	active or inactive	Status of the relationships to filter
depth	-	Maximum length of the relationship path search. Currently, not in use.
totalResults	true or false	View total records count
expand	all	View more details of the relationship
entityResource	-	More details of the entity whose relationships need to be retrieved. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".

Parameter	Description
totalResults	Value is set to the total number of relationships based on filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/entities/Patient?totalResults=true

Sample Request Body

```
{
"entityId": "1"
}
```

OR

```
{
"id": "0000000003"
}
```

OR

```
{
"attributes":[
{
"name":"FirstName",
"value":"Mark"
}
]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/relationships/4",
          "mediaType": "application/json",
          "rel": "canonical"
  }
],
"modifiedBy": null,
"modifiedDate": null,
"relationshipId": 4,
"source":
{
  "attributes": null,
  "entityId": 1,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/1",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
  {
    "attributes": null,
    "entityId": 22,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/22",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "status": null,
    "type": null
  },
  "type":
  {
    "attributes": null,
    "createdBy": null,
    "createdDate": null,
    "description": null,
    "direction": null,
    "effectiveDate": null,
    "end":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "status": "Active"
     },
     "expirationDate": null,
     "links":
     [
      {
       "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
       "mediaType": "application/json",
       "rel": "canonical"
      }
     ],
     "modifiedBy": null,
     "modifiedDate": null,
     "multiplicity": null,
     "name": "patient-of",
     "relationshipTypeId": 5,
     "start":
     {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
       {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
       }
      ],
      "name": "Patient",
      "status": "Active"
     },
     "status": "Active",
     "task": null
    },
    "validity": "Resolved"
   }
  ],
  "links":
   [
    {
     "href": "/ohmpi-rm/1.0.0/resources/relationships/Patient",
     "mediaType": "application/json",
     "rel": "canonical"
    }
   ],
   "totalResults": 1
}
```

Sample Response Code

200

5.5 POST /resources/relationships/{name}/{attribute}

This API is used to search relationships by using date attribute ranges and other criteria.

- If *relationship.management.text.index* is set as true in the ohmpi_rm.properties file, then the attribute field can only be createdDate, modifiedDate, expirationDate or effectiveDate.
- If *relationship.management.text.index* is set as false in the ohmpi_rm.properties file, then the attribute field can have custom date attributes as well as the one of the following: createdDate, modifiedDate, expirationDate or effectiveDate.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/{name}/{attribute}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name of relationships to be retrieved
attribute	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Attribute name of the relationship type to be searched using range search. It must be of type Date.
since	Should be in the format yyyy-MM-dd	Start date from which attribute must be searched. It is a mandatory parameter.
until	Should be in the format yyyy-MM-dd	End date till which attribute must be searched. Default is today's date.
status	active, inactive, or all	Status of relationships to filter
orderBy	Format: fieldname | sortOrder where, fieldname can be relationshipId, sourceId, targetId, createdBy, createdDate, modifiedBy, modifiedDate, expirationDate, effectiveDate, validity, or status; sortOrder can be asc or desc	Order by built-in attribute name. Default is the order by relationshipId in ascending order.
limit	Any positive integer	Maximum number of results to return in the REST API response. Default value is 1000.
offset	Any positive integer	Offset from which the results should be provided in the response. Default value is 1.
totalResults	true or false	View total records count
expand	all	View more details of the relationships
relationshipResource	-	Details of the relationships to be searched. For information on the fields for the relationship request body, see Section A.1.8, "Relationship Resource".

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".
totalResults	Value is set to the total number of relationships based on the filter. It might be greater than the number of results displayed in the REST API response, based on the offset and limit values. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/spouse-of/createdDate?since=2016-01-15&limit=1&offset=10

Sample Request Body

```
{
"attributes":[
{
"name":"DateOfMarriage",
"value":"03/17/1988"
}
]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/relationships/26",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "modifiedBy": null,
"modifiedDate": null,
"relationshipId": 26,
"source":
{
  "attributes": null,
  "entityId": 6,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/6",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 8,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/8",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  },
        "expirationDate": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "multiplicity": null,
        "name": "spouse-of",
        "relationshipTypeId": 7,
        "start":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/spouse-of/createdDate",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": null
}
```

Sample Response Code

200

5.6 POST /resources/relationships/{name}/{source}/{target}

This API is used to create a relationship by using source entity type, target entity type and relationship type.

Either entityId or id needs to be passed for the source and target entities in the request body.

If entityId is passed in the body, no new entity is created even if an entity with the entityId does not exist. If id is passed (along with or without other attributes) and the entity does not exist for the id, a new active entity is created based on the attributes passed in the body.

For creating a relationship, the relationship type name passed in the REST API is verified and the task id is found. If the task id is provided, corresponding task's groovy script is executed. If the task id is not provided, Default Relationship Update policy is used.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/{name}/{source}/{target}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name of relationship to be created
source	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Source Entity Type name
target	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Target Entity Type name
relationshipResource	-	Details of the source and target entities and the relationship to be created. For details on the fields for the relationship request body, see Section A.1.8, "Relationship Resource".

Response Parameters

Parameter	Description
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/patient-of/Patient/IndProvider

Sample Request Body

```
{
  "source":{
    "id": "200000002",
    "type": { "name": "IndProvider"  },
    "attributes": [
      {
        "name": "FirstName",
        "value": "Harry"
      }
    ]
  },
  "target": {
    "id": "0000001000",
    "type": { "name": "Patient"  }
  },
  "effectiveDate": "2016-01-14",
  "expirationDate": "2026-01-14",
  "type": {   "name": "patient-of"    },
  "attributes": [
    {
      "name": "Location",
      "value": "United States"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/41",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "relationshipId": 41,
  "source":
  {
    "attributes": null,
    "entityId": 81,
    "id": null,
    "links":
```

```
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/81",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 21,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/21",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  },
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/doctor-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "doctor-of",
    "relationshipTypeId": 21,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  },
  "validity": "Resolved"
}
```

Sample Response Code

200

5.7 POST /resources/relationships/entities/{entity}

This API is used to create a relationship from a single source or target entity by using the configured inferring rules.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/entities/{entity}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entity	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of the entity for which relationship is to be created

Parameter	Value	Description
entityResource	-	Details of the entity whose relationship needs to be created. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/entities/Patient

Sample Request Body

Assuming that you want to create a relationship of type primary-care-physician-of, the source entity type of the relationship type is Patient, and the target entity type is IndProvider. Also, an IndProvider entity having ID 0000000013 exists in system. Perform the following steps for the initial setup:

1. Create a ruleset with source as Patient and target as IndProvider.
2. Create a ruleset file for the ruleset and upload it using REST APIs or UI.

   For sample purpose, consider the ruleset with name primary-care-physician-creation and ruleset file defined in Section A.4.1, "Sample Ruleset File 1".
3. Create a task with the same name as defined in the ruleset file.
4. Create a groovy script for the task and upload it using REST APIs or UI.

   For sample purpose, consider the task with name PrimaryCarePhysicianCreationTask and the task groovy script defined in Section A.4.4, "Sample Task File 1".
5. Invoke the API using the following sample request body.

```
{
"id":"PA1301309",
"attributes":
  [
    {
      "name":"FirstName",
      "value":"John"
    },
    {
      "name":"LastName",
      "value":"D"
    },
    {
      "name":"Gender",
      "value":"M"
    },
    {
      "name":"City",
      "value":"CA"
    },
    {
      "name":"Provider.ID",
      "value":"0000000013"
    }
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/121",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "relationshipId": 121,
  "source":
  {
    "attributes": null,
    "entityId": 22,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/22",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "status": null,
    "type": null
  },
  "status": "Active",
  "target":
  {
    "attributes": null,
    "entityId": 201,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/201",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "status": null,
    "type": null
  },
  "type":
  {
    "attributes": null,
    "createdBy": null,
    "createdDate": null,
    "description": null,
    "direction": null,
    "effectiveDate": null,
    "end":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "expirationDate": null,
    "links":
    [
      {
        "href":
"/ohmpi-rm/1.0.0/metadata/relationships/primary-care-physician-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "primary-care-physician-of",
    "relationshipTypeId": 1,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
            "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "name": "IndProvider",
        "status": "Active"
      },
      "status": "Active",
      "task": null
    },
    "validity": "Resolved"
}
```

Sample Response Code

200

5.8 POST /resources/relationships/entities/{source}/{target}

This API is used to create relationships between the source entity and the target entity by using the configured inferring rules.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/entities/{source}/{target}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
source	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Source entity type name of the entity for which relationship is to be created
target	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Target entity type name of the entity for which relationship is to be created
entityResources	-	Details of the source and target entities whose relationship needs to be created. For details on the fields for the entity request body, see Section A.1.7, "Entity Resources".

Response Parameters

Parameter	Description
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/entities/Patient/IndProvider

Sample Request Body

Assuming that you want to create a relationship of type patient-of, the source entity type of the relationship type is Patient, and the target entity type is IndProvider. Perform the following steps for the initial setup:

1. Create a ruleset with source as Patient and target as IndProvider.
2. Create a ruleset file for the ruleset and upload it using REST APIs or UI.

   For sample purpose, consider the ruleset with name CreatePatientOfSampleRuleset and ruleset file defined in Section A.4.2, "Sample Ruleset File 2".
3. Create a task with the same name as defined in the ruleset file.
4. Create a groovy script for the task and upload it using REST APIs or UI.

   For sample purpose, consider the task with name CreatePatientOfSampleTask and the task groovy script defined in Section A.4.5, "Sample Task File 2".
5. Invoke the API using the following sample request body.

```
{
  "items" :
  [
    {
      "id":"PA130",
      "attributes":
      [
        {
          "name":"FirstName",
          "value":"John"
        },
        {
          "name":"LastName",
          "value":"D"
        },
        {
          "name":"Gender",
          "value":"M"
        },
        {
          "name":"City",
          "value":"CA"
        },
        {
          "name":"Provider.ID",
          "value":"0000000013"
        },
        {
          "name" : "Doctor",
          "value" : "Dr."
        }
      ]
    },
    {
      "id":"DA123",
      "attributes":
      [
        {
          "name":"FirstName",
          "value":"John"
        },
        {
          "name":"LastName",
          "value":"D"
        },
        {
          "name":"Gender",
          "value":"M"
        },
        {
          "name":"License",
          "value":"CA"
        },
        {
          "name":"Specialty",
          "value":"EAR"
        }
      ]
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/142",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "relationshipId": 142,
  "source":
  {
    "attributes": null,
  "entityId": 264,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/264",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 265,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/265",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "IndProvider",
    "status": "Active"
  },
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
    "mediaType": "application/json",
    "rel": "canonical"
   }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "multiplicity": null,
  "name": "patient-of",
  "relationshipTypeId": 5,
  "start":
  {
   "attributes": null,
   "description": null,
   "entityTypeId": null,
   "id": null,
   "links":
   [
    {
     "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
     "mediaType": "application/json",
     "rel": "canonical"
    }
   ],
   "name": "Patient",
   "status": "Active"
  },
  "status": "Active",
  "task": null
 },
 "validity": "Resolved"
}
```

Sample Response Code

200

5.9 PUT /resources/relationships/{id}

This API is used to update relationship by its ID.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/{id}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
id	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Relationship id of the relationship to be updated.
status	active or inactive	To activate or deactivate the relationship. *When status is passed, the body (relationshipResource) details will be ignored. But the body needs to be passed as {} (open curly brace and closed curly brace).*
relationshipResource	-	Details of the relationship to be updated. This will be considered for updating the relationship if the status parameter is not set to any value. For details on the fields for the relationship request body, see Section A.1.8, "Relationship Resource".

Response Parameters

Parameter	Description
relationship	For information on the relationship type attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use theGET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid resource id
500	Error while processing

5.9.1 To Update Relationship Status

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/1?status=active

Sample Request Body

```
{}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": "ohmpi",
  "createdDate": "2016-01-14 15:03:20",
  "effectiveDate": "2016-01-14 15:03:19",
  "expirationDate": "2026-01-14 15:03:19",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/1",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"modifiedBy": "ohmpi",
"modifiedDate": "2016-01-14 15:03:20",
"relationshipId": 1,
"source":
{
  "attributes": null,
  "entityId": 21,
  "id": null,
  "links":
  [
  {
    "href": "/ohmpi-rm/1.0.0/resources/entities/21",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"status": null,
"type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": "ohmpi",
  "createdDate": "2016-01-14 14:45:49",
  "description": null,
  "direction": "BIDIRECTIONAL",
  "effectiveDate": "2016-01-14 14:45:49",
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "expirationDate": "2026-01-14 14:45:49",
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": "ohmpi",
    "modifiedDate": "2016-01-14 14:45:49",
    "multiplicity": "MANYTOMANY",
    "name": "patient-of",
    "relationshipTypeId": 1,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  },
  "validity": "Resolved"
}
```

Sample Response Code

200

5.9.2 To Update the Relationship Attributes and Other Details

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/41

Sample Request Body

```
{
  "expirationDate":"2030-01-14 00:00:00",
  "attributes":
```

```
  [
    {
      "name": "Location",
      "value": "California"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes":
  [
    {
      "name": "Location",
      "relationship":
      {
        "attributes": null,
        "createdBy": null,
        "createdDate": null,
        "effectiveDate": null,
        "expirationDate": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/relationships/41",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "relationshipId": 41,
        "source": null,
        "status": null,
        "target": null,
        "type": null,
        "validity": null
      },
      "relationshipAttributeId": 1,
      "value": "California"
    }
  ],
  "createdBy": "ohmpi",
  "createdDate": "2016-01-18 13:31:18",
  "effectiveDate": "2016-01-18 13:31:17",
  "expirationDate": "2030-01-14 00:00:00",
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/41",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": "ohmpi",
  "modifiedDate": "2016-01-18 14:20:04",
  "relationshipId": 41,
  "source":
```

```
{
  "attributes": null,
  "entityId": 81,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/81",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 21,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/21",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes":
  [
    {
      "attributeTypeId": 42,
      "defaultValue": null,
      "description": null,
      "indexed": "Y",
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/relationships/doctor-of",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "mandatory": "N",
      "maxValue": null,
      "minValue": null,
      "name": "Location",
      "readOnly": "N",
      "sizeValue": 40,
      "status": "Active",
      "type": "String"
    }
  ],
  "createdBy": "ohmpi",
    "createdDate": "2016-01-18 13:28:17",
    "description": "doctor of relationship",
    "direction": "BIDIRECTIONAL",
    "effectiveDate": "2016-01-18 13:28:17",
    "end":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "expirationDate": "2026-01-18 13:28:17",
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/relationships/doctor-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": "ohmpi",
    "modifiedDate": "2016-01-18 13:28:17",
    "multiplicity": "MANYTOMANY",
    "name": "doctor-of",
    "relationshipTypeId": 21,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  },
  "validity": "Resolved"
}
```

Sample Response Code

200

5.10 PUT /resources/relationships/entities/{name}

This API is used to synchronize the relationships with ADD, UPD, MRG, UMRG, DEA, and REA events.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/resources/relationships/entities/{name}

Request Content Type

application/json

Request Type

HTTP PUT

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Name of the entity type to be synchronized from RM Agent
event	ADD, UPD, MRG, UMRG, DEA, or REA	Event type to be used for synchronization
totalResults	true or false	View total records count
expand	all	View more details of the relationships
entityResources	-	Details of the entities that need to be synchronized. At least one entity must be passed and will be considered as the source entity. If second entity is passed, it will be considered as the target entity. For details on the fields for the entity request body, see Section A.1.7, "Entity Resources".

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure".
totalResults	Value is set to the total number of relationships based on the filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid resource name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/resources/relationships/entities/Patient?event=ADD

Sample Request Body

Assuming that you want to create a relationship of type primary-care-physician-of, the source entity type of the relationship type is Patient, and the target entity type is IndProvider. Also, an IndProvider entity having ID CA786 exists in the RM database. Perform the following steps for the initial setup:

1. Create a domain with the name Patient using REST APIs or UI.
2. Create a ruleset with source as Patient and target as IndProvider.
3. Create a ruleset file for the ruleset and upload it using REST APIs or UI.

   For sample purpose, consider the ruleset with name primary-care-physician-creation and ruleset file defined in Section A.4.1, "Sample Ruleset File 1".
4. Create a task with the same name as defined in the ruleset file.
5. Create a groovy script for the task and upload it using REST APIs or UI.

   For sample purpose, consider the task with name PrimaryCarePhysicianCreationTask and the task groovy script defined in Section A.4.4, "Sample Task File 1".
6. Generate a wallet file and deploy the MPI Agent for the Patient MPI application. For information, see *Oracle Healthcare Master Person Index Relationship Management User's Guide*.
7. Create a new Patient in the MPI database, either using MIDM or MPI SOAP APIs.
8. Invoke the RM API using the following sample request body.

```
{
  "items":
  [
    {
      "id":"CA1002",
      "attributes":
      [
        {
          "name":"FirstName",
          "value":"David"
        },
        {
          "name":"LastName",
          "value":"K"
        },
        {
          "name":"Gender",
          "value":"M"
        },
        {
          "name":"City",
          "value":"CA"
        },
        {
          "name":"Provider.ID",
          "value":"CA786"
        }
      ]
    }
  ]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/relationships/122",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "modifiedBy": null,
      "modifiedDate": null,
      "relationshipId": 122,
      "source":
      {
        "attributes": null,
        "entityId": 203,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/entities/203",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "status": null,
        "type": null
      },
      "status": "Active",
      "target":
      {
        "attributes": null,
        "entityId": 206,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/entities/206",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "status": null,
        "type": null
      },
      "type":
      {
        "attributes": null,
        "createdBy": null,
        "createdDate": null,
        "description": null,
        "direction": null,
        "effectiveDate": null,
        "end":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "expirationDate": null,
        "links":
        [
          {
            "href":
"/ohmpi-rm/1.0.0/metadata/relationships/primary-care-physician-of",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "multiplicity": null,
        "name": "primary-care-physician-of",
        "relationshipTypeId": 1,
        "start":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "IndProvider",
          "status": "Active"
        },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/entities/Patient",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": null
}
```

Sample Response Code

200

6

Integration Resources

The integration resources are used to integrate with the underlying OHMPI applications. The integration resources contain the following APIs:

- GET /integration
- POST /integration/relationships/{name}
- POST /integration/relationships/{entity}
- POST /integration/relationships/{name}/{source}/{target}
- POST /integration/relationships/entities/{entity}
- POST /integration/relationships/entities/{source}/{target}

> **Note:** Before using the integration APIs, make sure that the domains are configured correctly and the MPI application is up and running. The name of the OHMPI domain name or the OHMPI primary object must exactly match to the name of the RM entity type.

6.1 GET /integration

This API is used to get integration REST API versioning information.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
version	OHMPI RM version number
description	The description of the integration resources
url	The URL of this REST API

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration

Sample Response Body

```
{
"version": "1.0.0",
"description": "OHMPI RM MPI Integration Resources",
"url": http://localhost:7001/ohmpi-rm/1.0.0/integration
}
```

Sample Response Code

200

6.2 POST /integration/relationships/{name}

This API is used to search relationships between the source MPI domain and the target MPI domain with the relationship type filter.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration/relationships/{name}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name of relationship to be searched
source	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Source Entity Type name
target	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Target Entity Type name
totalResults	true or false	View total records count
expand	all	View more details of the relationships

Parameter	Value	Description
relationshipResource	-	Details of the relationships to be searched. For details of the fields for relationship request body, see Section A.1.9, "Integration Relationship Resource".

Response Parameters

Parameter	Description
relationship	For information on relationship attributes in response, see Section A.2.7, "Relationship Response Structure".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration/relationships/spouse-of?source=Patient&target=Patient

Sample Request Body

```
{
  "source": {
  "type":
    {
      name": "Patient"
    },
  "attributes": [
    {
      "name": "FirstName",
      "value": "Akshay"
    },
    {
      "name": "LastName",
      "value": "Kumar"
    }
  ]
},
  "target": {
    "type":
      {
        "name": "Patient"
      },
    "attributes": [
      {
      "name": "FirstName",
      "value": "Balaji"
      },
      {
```

```
      "name": "LastName",
      "value": "Kadam"
      }
    ]
  }
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/resources/relationships/45",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "modifiedBy": null,
      "modifiedDate": null,
      "relationshipId": 45,
      "source":
      {
        "attributes": null,
        "entityId": 91,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/entities/91",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "status": null,
        "type": null
      },
      "status": "Active",
      "target":
      {
        "attributes": null,
        "entityId": 92,
        "id": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/resources/entities/92",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  },
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "multiplicity": null,
  "name": "spouse-of",
  "relationshipTypeId": 22,
  "start":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
    "status": "Active"
  },
  "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/integration/relationships/spouse-of",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": null
}
```

Sample Response Code

200

6.3 POST /integration/relationships/{entity}

This API is used to search relationships from the source MPI domain or target MPI domain and with the relationship type filter.

Difference between the POST /integration/relationships/{entity} API and the POST /resources/relationships/{entity} API is that the POST /integration/relationships/{entity} API queries the MPI database of the corresponding domain and retrieves the EUID from that database. Whereas, the POST /resources/relationships/{entity} API queries the RM database directly.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration/relationships/{entity}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entity	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of entity whose relationships need to be retrieved
label	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name whose relationships need to be retrieved. It is a mandatory parameter.
depth	-	Maximum length of the relationship path search. Currently, not in use.
totalResults	true or false	View total records count
expand	all	View more details of the relationship

Parameter	Value	Description
entityResource	-	More details of the entity whose relationships need to be retrieved. For details on the fields for entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
items	An array containing relationships
relationship	For information on relationship attributes in response, see Section A.2.7, "Relationship Response Structure".
totalResults	Value is set to the total number of relationships based on filter. The value is displayed if totalResults is set to true in the request parameter. Otherwise, the value is null.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration/relationships/Patient?label=spouse-of

Sample Request Body

```
{
  "attributes": [
    {
      "name": "FirstName",
      "value": "Peter"
    }
  ]
}
```

Sample Response Body

```
{
  "items":
  [
    {
      "attributes": null,
      "createdBy": null,
      "createdDate": null,
      "effectiveDate": null,
      "expirationDate": null,
      "links":
      [
        {
```

```
    "href": "/ohmpi-rm/1.0.0/resources/relationships/42",
    "mediaType": "application/json",
    "rel": "canonical"
  }
],
"modifiedBy": null,
"modifiedDate": null,
"relationshipId": 42,
"source":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 82,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/82",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "expirationDate": null,
        "links":
        [
          {
            "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
            "mediaType": "application/json",
            "rel": "canonical"
          }
        ],
        "modifiedBy": null,
        "modifiedDate": null,
        "multiplicity": null,
        "name": "spouse-of",
        "relationshipTypeId": 22,
        "start":
        {
          "attributes": null,
          "description": null,
          "entityTypeId": null,
          "id": null,
          "links":
          [
            {
              "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
              "mediaType": "application/json",
              "rel": "canonical"
            }
          ],
          "name": "Patient",
          "status": "Active"
        },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
    }
  ],
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/integration/relationships/Patient",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalResults": null
}
```

Sample Response Code

200

6.4 POST /integration/relationships/{name}/{source}/{target}

This API is used to create a relationship between the source MPI domain, target MPI domain, and relationship type.

Differences between the POST /integration/relationships/{name}/{source}/{target} API and the POST /resources/relationships/{name}/{source}/{target} API is that:

- In the POST /integration/relationships/{name}/{source}/{target} API, the MPI database of the corresponding entity type is queried to check whether or not the record exists. Accordingly, create or retrieve an entry of the MPI database, and then entities and relationship are created in the RM database.
- In the POST /resources/relationships/{name}/{source}/{target} API, entities and relationships are directly created in the RM database. There is no interaction with the MPI database.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration/relationships/{name}/{source}/{target}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
name	Matching the regex [a-zA-Z][a-zA-Z_0-9-]+	Relationship type name of relationship to be created
source	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Source Entity Type name
target	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Target Entity Type name
relationshipResource	-	Details of the relationships to be searched. For details on the fields for relationship request body, see Section A.1.9, "Integration Relationship Resource". Any date values of the source or target entity must be provided in the MPI date format of the corresponding entity type or domain.

Response Parameters

Parameter	Description
relationship	For information on relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration/relationships/spouse-of/Patient/Patient

Sample Request Body

Assume that the date format of Patient MPI is yyyy/MM/dd and date format of RM is yyyy-MM-dd HH:mm:ss.

```
{
  "source": {
    "type":
      {
        "name": "Patient"
      },
    "attributes": [
      {
      "name": "LocalId",
      "value": "2000099333"
      },
      {
      "name": "SystemCode",
      "value": "BBB"
      },
      {
      "name": "FirstName",
      "value": "Peter"
      },
      {
      "name": "LastName",
      "value": "Wills"
      },
      {
      "name": "Gender",
      "value": "M"
      },
      {
      "name": "SSN",
      "value": "994936919"
      },
      {
      "name": "DOB",
      "value": "1987/01/03"
      }
    ]
  },
  "target": {
    "type":
      {
      "name": "Patient"
      },
    "attributes": [
      {
      "name": "LocalId",
      "value": "1000023351"
      },
      {
      "name": "SystemCode",
      "value": "BBB"
      },
      {
      "name": "FirstName",
      "value": "Mary"
      },
      {
      "name": "LastName",
      "value": "John"
      },
      {
      "name": "Gender",
      "value": "F"
      },
      {
      "name": "SSN",
      "value": "12761813"
      },
      {
      "name": "DOB",
      "value": "1988/02/01"
      }
    ]
  },
  "type":
    {
      "name": "spouse-of"
    },
  "attributes": [
    {
      "name": "DateOfMarriage",
      "value": "2015-02-23 20:10:10"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/61",
      "mediaType": "application/json",
      "rel": "canonical"
    }
```

```
],
"modifiedBy": null,
"modifiedDate": null,
"relationshipId": 61,
"source":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 101,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/101",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "name": "Patient",
      "status": "Active"
    },
    "expirationDate": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/relationships/spouse-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "spouse-of",
    "relationshipTypeId": 22,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  },
  "validity": "Resolved"
}
```

Sample Response Code

200

6.5 POST /integration/relationships/entities/{entity}

This API is used to create a relationship from a single source or target MPI domain by using the configured inferring rules.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration/relationships/entities/{entity}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
entity	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Entity type name of the entity for which relationship is to be created
entityResource	-	Details of the entity whose relationship needs to be created. For details on the fields for the entity request body, see Section A.1.6, "Entity Resource".

Response Parameters

Parameter	Description
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration/relationships/entities/Patient

Sample Request Body

Assuming that you want to create a relationship of type primary-care-physician-of, the source entity type of the relationship type is Patient, and the target entity type is IndProvider. Also, an IndProvider entity having ID 0000000013 exists in system. Perform the following steps for the initial setup:

1. Create a ruleset with source as Patient and target as IndProvider.
2. Create a ruleset file for the ruleset and upload it using REST APIs or UI.

   For sample purpose, consider the ruleset with name primary-care-physician-creation and ruleset file defined in Section A.4.1, "Sample Ruleset File 1".
3. Create a task with the same name as defined in the ruleset file.
4. Create a groovy script for the task and upload it using REST APIs or UI.

   For sample purpose, consider the task with name PrimaryCarePhysicianCreationTask and the task groovy script defined in Section A.4.4, "Sample Task File 1".
5. Invoke the API using the following sample request body.

```
{
  "id":"PA1311",
  "attributes":
  [
    {
      "name" : "LocalId",
      "value" : "1000000000"
    },
    {
      "name" : "SystemCode",
      "value" : "Oracle"
    },
    {
      "name":"FirstName",
      "value":"John"
    },
    {
      "name":"LastName",
      "value":"D"
    },
    {
      "name":"Gender",
      "value":"M"
    },
    {
      "name":"City",
      "value":"CA"
    },
    {
      "name":"Provider.ID",
      "value":"0000000013"
    },
    {
      "name":"DOB",
      "value":"2015/01/02"
    },
    {
      "name" : "SSN",
      "value" : "111111111"
    },
    {
      "name" : "Provider.Name",
      "value" : "Appolo"
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/143",
      "mediaType": "application/json",
    "rel": "canonical"
  }
],
"modifiedBy": null,
"modifiedDate": null,
"relationshipId": 143,
"source":
{
  "attributes": null,
  "entityId": 22,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/22",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"status": "Active",
"target":
{
  "attributes": null,
  "entityId": 275,
  "id": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/entities/275",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "status": null,
  "type": null
},
"type":
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "description": null,
  "direction": null,
  "effectiveDate": null,
  "end":
  {
    "attributes": null,
    "description": null,
    "entityTypeId": null,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
        "mediaType": "application/json",
        "rel": "canonical"
      }
      ],
      "name": "Patient",
      "status": "Active"
    },
    "expirationDate": null,
    "links":
    [
      {
        "href":
"/ohmpi-rm/1.0.0/metadata/relationships/primary-care-physician-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "primary-care-physician-of",
    "relationshipTypeId": 1,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "status": "Active"
    },
    "status": "Active",
    "task": null
  },
  "validity": "Resolved"
}
```

Sample Response Code

200

6.6 POST /integration/relationships/entities/{source}/{target}

This API is used to create a relationship between the source and target MPI domain by using the configured inferring rules.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/integration/relationships/entities/{source}/{target}

Request Content Type

application/json

Request Type

HTTP POST

Response Content Type

application/json;charset=UTF-8

Request Query Parameters

Parameter	Value	Description
source	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Source entity type name of the entity for which relationship is to be created
target	Matching the regex [a-zA-Z][a-zA-Z_0-9]+	Target entity type name of the entity for which relationship is to be created
entityResources	-	Details of the source and target entities whose relationship needs to be created. For details on the fields for the entity request body, see Section A.1.7, "Entity Resources".

Response Parameters

Parameter	Description
relationship	For information on the relationship attributes in response, see Section A.2.7, "Relationship Response Structure". To display all the attributes of the relationship type, use the GET /resources/relationships/{id} API with expand=all.

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
400	Invalid argument/Invalid Resource Name
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/integration/relationships/entities/Patient/IndProvider

Sample Request Body

Assuming that you want to create a relationship of type patient-of, the source entity domain of the relationship type is Patient, and the target entity domain. Perform the following steps for the initial setup:

1. Create a ruleset with source as Patient and target as IndProvider.
2. Create a ruleset file for the ruleset and upload it using REST APIs or UI.

   For sample purpose, consider the ruleset with name CreatePatientOfSampleRuleset and ruleset file defined in Section A.4.3, "Sample Ruleset File 3".
3. Create a task with the same name as defined in the ruleset file.
4. Create a groovy script for the task and upload it using REST APIs or UI.

For sample purpose, consider the task with name CreatePatientOfSampleTask and the task groovy script defined in Section A.4.5, "Sample Task File 2".

5. Invoke the API using the following sample request body.

```
{
  "items" :
  [
    {
      "id":"PA1311",
      "attributes":
      [
        {
          "name" : "LocalId",
          "value" : "1000000001"
        },
        {
          "name" : "SystemCode",
          "value" : "Oracle"
        },
        {
          "name":"FirstName",
          "value":"Ricky"
        },
        {
          "name":"LastName",
          "value":"D"
        },
        {
          "name":"Gender",
          "value":"M"
        },
        {
          "name":"City",
          "value":"CA"
        },
        {
          "name":"Provider.ID",
          "value":"0000000014"
        },
        {
          "name":"DOB",
          "value":"2015/01/02"
        },
        {
          "name" : "Doctor",
          "value" : "Dr."
        },
        {
          "name" : "SSN",
          "value" : "111111111"
        },
        {
          "name" : "Provider.Name" ,
          "value" :"Nancy"
        }
      ]
    },
    {
      "id":"DA12312",
      "attributes":
```

```
[
  {
    "name" : "LocalId",
    "value" : "0000012299"
  },
  {
    "name" : "SystemCode",
    "value" : "Oracle"
  },
  {
    "name":"FirstName",
    "value":"Johny"
  },
  {
    "name":"LastName",
    "value":"Perch"
  },
  {
    "name":"Gender",
    "value":"F"
  },
  {
    "name":"License",
    "value":"CA"
  },
  {
    "name":"DOB",
    "value":"2015/01/02"
  },
  {
    "name":"Specialty",
    "value":"Nutritionist"
  },
  {
    "name" : "ProviderStatus",
    "value" : "ACTIVE"
  },
  {
    "name" : "Names.NameType",
    "value" : "RN"
  },
  {
    "name" : "Names.FirstName",
    "value" : "Ricky"
  },
  {
    "name" : "Names.LastName",
    "value" : "B"
  },
  {
    "name" : "Locations.LocationType",
    "value" : "M1"
  },
  {
    "name" : "Locations.AddressLine1",
    "value" : "lane10"
  },
  {
    "name" : "Locations.City",
    "value" : "CA"
```

```
        }
      ]
    }
  ]
}
```

Sample Response Body

```
{
  "attributes": null,
  "createdBy": null,
  "createdDate": null,
  "effectiveDate": null,
  "expirationDate": null,
  "links":
  [
    {
      "href": "/ohmpi-rm/1.0.0/resources/relationships/144",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "modifiedBy": null,
  "modifiedDate": null,
  "relationshipId": 144,
  "source":
  {
    "attributes": null,
    "entityId": 282,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/282",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "status": null,
    "type": null
  },
  "status": "Active",
  "target":
  {
    "attributes": null,
    "entityId": 283,
    "id": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/resources/entities/283",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "status": null,
    "type": null
  },
  "type":
```

```
  {
    "attributes": null,
    "createdBy": null,
    "createdDate": null,
    "description": null,
    "direction": null,
    "effectiveDate": null,
    "end":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/IndProvider",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "IndProvider",
      "status": "Active"
    },
    "expirationDate": null,
    "links":
    [
      {
        "href": "/ohmpi-rm/1.0.0/metadata/relationships/patient-of",
        "mediaType": "application/json",
        "rel": "canonical"
      }
    ],
    "modifiedBy": null,
    "modifiedDate": null,
    "multiplicity": null,
    "name": "patient-of",
    "relationshipTypeId": 5,
    "start":
    {
      "attributes": null,
      "description": null,
      "entityTypeId": null,
      "id": null,
      "links":
      [
        {
          "href": "/ohmpi-rm/1.0.0/metadata/entities/Patient",
          "mediaType": "application/json",
          "rel": "canonical"
        }
      ],
      "name": "Patient",
      "status": "Active"
      },
        "status": "Active",
        "task": null
      },
      "validity": "Resolved"
}
```

Sample Response Code

200

7

Stats Resources

The stats resources contain the following APIs:

- GET /stats
- GET /stats/metadata
- GET /stats/resources

7.1 GET /stats

This API is used to get stats REST API versioning information.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/stats

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
version	OHMPI RM version number
description	The description of stats resources
url	The URL of this REST API

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/stats

Sample Response Body

```
{
  "version": "1.0.0",
  "description": "OHMPI RM Stats Resources",
  "url": "http://localhost:7001/ohmpi-rm/1.0.0/stats"
}
```

Sample Response Code

200

7.2 GET /stats/metadata

This API is used to get metadata stats information, such as active and inactive count of entity types, relationship types, rulesets, tasks, and domains.

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/stats/metadata

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
totalActiveEntityTypes	Number of Active Entity Types in RM
totalInactiveEntityTypes	Number of Inactive Entity Types in RM
totalActiveRelationshipTypes	Number of Active Relationship Types in RM
totalInactiveRelationshipTypes	Number of Inactive Relationship Types in RM
totalActiveRulesets	Number of Active Rulesets Types in RM
totalInactiveRulesets	Number of Inactive Rulesets Types in RM
totalActiveTasks	Number of Active Tasks in RM
totalInactiveTasks	Number of Inactive Tasks in RM
totalDomains	Number of Domains in RM
links	Provides information of REST API invoked. For more information, see Section A.3, "Links".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/stats/metadata

Sample Response Body

```
{
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/stats/metadata",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalActiveEntityTypes": 6,
  "totalActiveRelationshipTypes": 10,
  "totalActiveRulesets": 1,
  "totalActiveTasks": 1,
  "totalDomains": 3,
  "totalInactiveEntityTypes": 0,
  "totalInactiveRelationshipTypes": 0,
  "totalInactiveRulesets": 0,
  "totalInactiveTasks": 0
}
```

Sample Response Code

200

7.3 GET /stats/resources

This API is used to get resources stats information, such as active and inactive count of entity, relationship, potential or resolved relationships, and transactions (last 30 days).

Request URI

http(s)://<server>:<port>/ohmpi-rm/1.0.0/stats/resources

Request Type

HTTP GET

Response Content Type

application/json;charset=UTF-8

Response Parameters

Parameter	Description
totalActiveEntities	Number of Active Entities in RM
totalInactiveEntities	Number of Inactive Entities in RM
totalActiveRelationships	Number of Active Relationships in RM
totalInactiveRelationships	Number of Inactive Relationships in RM
totalPotentialRelationships	Number of Relationships with validity=Potential in RM
totalResolvedRelationships	Number of Relationships with validity=Resolved in RM
totalTransactionsLastMonth	Number of Transactions (Create, Update, Activate, or Deactivate) in last 30 days

Parameter	Description
links	Provides information of REST API invoked. For more information, see Section A.3, "Links".

Response Status Code

Status Code	Description
200	Success
404	Incorrect URI
500	Error while processing

Sample Request URI

http://localhost:7001/ohmpi-rm/1.0.0/stats/resources

Sample Response Body

```
{
  "links": [
    {
      "href": "/ohmpi-rm/1.0.0/stats/resources",
      "mediaType": "application/json",
      "rel": "canonical"
    }
  ],
  "totalActiveEntities": 31,
  "totalActiveRelationships": 85,
  "totalInactiveEntities": 20,
  "totalInactiveRelationships": 32,
  "totalPotentialRelationships": 37,
  "totalResolvedRelationships": 80,
  "totalTransactionsLastMonth": 184
}
```

Sample Response Code

200

8 Secure Development

This section provides guidelines and recommendations for using and developing against the REST Web Services APIs published by OHMPI RM. The following sections includes the recommended practices for secure usage of the RM APIs.

8.1 Roles and Permissions

A developer working with the REST APIs is subject to the same roles and permissions required to use OHMPI RM applications. To work with the REST APIs, developers must have the access permission of the RESTful Web Services.

8.2 Valid Content Types

The REST API supports application/XML and application/JSON. The RM REST APIs support application/JSON content type only. Developers must use the correct MIME type in the Accept and Content-Type HTTP headers. Otherwise, the API will reject the requests.

8.3 Injection

To prevent SQL injections, the REST API uses bind variables in SQL queries that might involve any user input.

The data that the REST API returns is properly encoded for their returned types in the JSON format. Developers should encode data based on the context they are using. The REST API does not know the context, so safeguards are important.

8.4 Broken Authentication and Session Management

The REST API is stateless and, therefore, the RM service does not maintain an HTTP session information and status for the duration of multiple requests. Therefore, the typical security issues associated with HTTP stateful sessions do not apply.

However, the API does require transmission of the username and password in the HTTP authorization header. Developers, therefore, should ensure that the username and password are stored securely and use the HTTPS protocol. For example, this might be the Oracle Platform Security Services (OPSS) framework.

8.5 Cross-Site Scripting

The API validates input from developers before using it. For outbound data from the REST API, developers are responsible for properly encoding the data, depending on the context where the data is being used.

8.6 Insecure Direct Object References

The REST API ensures that the invoker has the permissions to access every requested resource and representation or entity returned.

8.7 Sensitive Data Exposure

The username and password are the only business-agnostic data that is considered sensitive for the REST API. The data is transmitted over HTTPS and is never at rest; that is, it is not stored or output to log files. Developers should also provide safe and secure storage when data is at rest.

8.8 Missing Function-Level Access Control

The REST API ensures that the invoker has the permissions to access every requested resource and representation or entity returned.

8.9 Cross-Site Request Forgery

Because the REST API is stateless, there is no HTTP session to exploit via Cross-Site Request Forgery (CSRF). Every request needs to carry the Authorization header with valid values; otherwise, it is rejected.

8.10 Using Known Vulnerable Components

The REST API stack is constantly updated with the latest security fixes and patches. Oracle recommends that developers using the API do the same.

8.11 Non-validated Redirects and Forwards

Although REST API responses might provide URLs to other resources and/or sub-resources in the representations that are returned, none of these URLs contain any user input. Therefore, the REST API is not subject to this vulnerability.

A

References

A.1 Request Structure

A.1.1 Entity Type Resource

Use this request structure in the API body when one of the request parameters is set as entityTypeResource.

The following are the parameters:

- **name** (mandatory): Name of the entity type to be created, updated, or searched (depending on the API).
- **description**: Description of the entity type.
- **id**: Describes the details that are stored for the entity resource in this field. For example, EUID. This is required while creating a new entity type, otherwise optional.
- **status**: Status of the entity type: Active or Inactive. This value is provided to activate or deactivate an entity type. While creating a new entity type, the value is not considered and always an Active entity type is created. For updating an entity type, use the PUT /metadata/entities/{name} API must be used. For information, see Section 2.5, "PUT /metadata/entities/{name}". Status can be passed as request parameter (recommended) or in the body of the API.
- **attributes**: An array of attribute types for the entity type.

  Each attribute type contains the following fields:

  - **name** (mandatory): Name of the attribute type. It must be unique for the entity type.
  - **description**: Description of the attribute type.
  - **type**: Defines the data type of the attribute type to be created. Valid options are String, Char, Integer, Float, Double, Date, Timestamp, Boolean, and Object.

    This is a mandatory field while creating new attributetype, otherwise optional. The data type of an attribute cannot be modified once set.
- **sizeValue**: Size of the String data type attribute type. For attributes of other datatype, the size is determined by the system and you cannot pass the value.

- **defaultValue**: Default value of the attribute type to be used while creating entities, if the attribute is not mandatory. Value for the attribute is not passed while creating an entity.
- **minValue**: Minimum value that the attribute type can take. This is used for Integer, Float, Double, Date, and Timestamp datatypes.
- **maxValue**: Maximum value that the attribute type can take. This is used for Integer, Float, Double, Date, and Timestamp datatypes.
- **mandatory**: Set Y if the attribute type must have a value when an entity is created, otherwise set N. If you do not set a value, system sets the default value as N.
- **readOnly**: Set Y if the attribute type can be set only when an entity is created, otherwise set N. If you do not set a value, system sets the default value as N.
- **indexed**: Set Y if the attribute type is used to do a search for the entities, otherwise set N. If you do not set a value, system sets the default value as Y.
- **status**: Status of attribute type: Active or Inactive. This value is used to update the status of the attribute. While creating an attribute type, its status is always active. You can later set the status as Inactive to deactivate the attribute type. Alternately, use the PUT /metadata/entities/{name}/{attribute} API to activate or deactivate the attribute type. For information, see Section 2.6, "PUT /metadata/entities/{name}/{attribute}".

A.1.2 Relationship Type Resource

Use this request structure in the API body when one of the request parameters is set as relationshipTypeResource.

The following are the parameters:

- **name** (mandatory): Name of the relationship type to be created, updated, or searched (depending on the API).
- **description**: Description of the relationship type.
- **start**: Details of the source entity type. This is required while creating a new relationship type.
  - **name**: Name is a mandatory field for the start parameter.
- **end**: Details of the target entity type. This is required while creating new relationship type.
  - **name**: Name is a mandatory field for the end parameter.
- **status**: Status of relationship type: Active or Inactive. This value is provided to activate or deactivate a relationship type. While creating a new relationship type, the value is not considered and always an Active relationship type is created. For updating a relationship type, the PUT /metadata/relationships/{name} API is used. For information, see Section 2.10, "PUT /metadata/relationships/{name}". Status can be passed as a request parameter (recommended) or in the body of the API.
- **direction**: Determines the direction in which the relationship type is created between the source and target. Possible values are UNIDIRECTIONAL and BIDIRECTIONAL. If you do not provide any value, BIDIRECTIONAL is set.
- **multiplicity**: Determines the multiplicity of the relationships for the relationship type. Possible values are ONETOONE, ONETOMANY, and MANYTOMANY.

Currently, this is not used. If you do not provide any value, MANYTOMANY is set.

- **createdBy**: Determines who created the relationship type. System will set this value to the user's id who is logged on to the REST API, only when relationship type is created. The value provided by you will be ignored. You cannot update the value later.
- **modifiedBy**: Determines who last modified the relationship type. System will set this value to the user's id who is logged on to the REST API, whenever a relationship type is created or modified. The value provided by you will be ignored.
- **createdDate**: Determines when the relationship type was created. System will set this value to current system date, only when a relationship type is created. The value provided by you will be ignored. You cannot update the value later.
- **modifiedDate**: Determines when the relationship type was last modified. System will set this value to current system date, when a relationship type is created or modified. The value provided by you will be ignored.
- **effectiveDate**: You can provide a value for this and this value needs to be passed in the date format *relationship.management.date.format* defined in the ohmpi_rm.properties file. If you do not provide a value, it will be set to current system date when the relationship type is created.
- **expirationDate**: Determines when the relationship type expires, so that no new relationships of that type can be created. You can provide a value for this and this value needs to be passed in the date format *relationship.management.date.format* defined in the ohmpi_rm.properties file. If you do not provide a value, it will be set to current system date plus 10 years from when the relationship type is created.
- **task**: Determines the task script to be invoked when a create relationship API is invoked. If not provided, default relationship update policy is used. For information on the relationship update policy, see *Oracle Healthcare Master Person Index Relationship Management User's Guide*.
  - **name**: Indicates the task name, which must match the name from the rm_task table.
- **attributes**: An array of attribute types for the relationship type.

  Each attribute type contains the following fields:

  - **name** (mandatory): Name of the attribute type. It must be unique for the relationship type.
  - **description**: Description of the attribute type.
  - **type**: Defines the data type of the attribute type to be created. Valid options are String, Char, Integer, Float, Double, Date, Timestamp, Boolean, and Object.

    This is a mandatory field while creating new attribute type, otherwise optional. The data type of an attribute cannot be modified once set.
  - **sizeValue**: Size of the String data type attribute type. For attributes of other datatype, the size is determined by the system and you cannot pass the value.
  - **defaultValue**: Default value of the attribute type to be used while creating relationships, if the attribute is not mandatory and the value for the attribute is not passed while creating an relationship.
  - **minValue**: Minimum value that the attribute type can take. It is used for Integer, Float, Double, Date, and Timestamp datatypes.

- **maxValue**: Maximum value that the attribute type can take. It is used for Integer, Float, Double, Date, and Timestamp datatypes.
- **mandatory**: Set Y if the attribute type must have a value when a relationship is created, otherwise set N. If you do not set a value, the system sets the default value as N.
- **readOnly**: Set Y if the attribute type can be set only when a relationship is created, otherwise set N. If you do not set a value, the system sets the default value as N.
- **indexed**: Set Y if the attribute type is used to do a search for the relationships, otherwise set N. If you do not set a value, the system sets the default value as Y.
- **status**: Status of attribute type: Active or Inactive. This value is used to update the status of the attribute. While creating an attribute type, its status is always active. You can later set the status as Inactive to deactivate the attribute type. Alternately, use the PUT /metadata/relationships/{name}/{attribute} API to activate or deactivate the attribute type. For information, see Section 2.11, "PUT /metadata/relationships/{name}/{attribute}".

A.1.3 Domain Resource

Use this request structure in the API body when one of the request parameters is set as domainResource.

The following are the parameters:

- **name** (mandatory): Name of the domain to be registered or updated. To register a domain, the entity type of the same name must exist in the RM database.
- **url**: URL of the MPI application for the domain. It must be in the following format:

  ```
  (t3|iiop)(s)?://[a-zA-Z0-9][a-zA-Z0-9-.]*:([0-9]{4})
  ```
  For example, t3://localhost:7001.
- **jndiname**: JNDI name of the MPI Application. It must be in the following format:

  ```
  (ejb)/[a-zA-Z0-9]+
  ```
  For example, ejb/PatientMasterController.
- **userid**: Username of the user to which the MasterIndex.Admin group is assigned.
- **usercode**: Password of the user in the base64 encoded format.

A.1.4 RmRuleset Resource

Use this request structure in the API body when one of the request parameters is set as rmRuleSetResource.

The following are the parameters:

- **name** (mandatory): Name of the ruleset to be registered or updated.
- **description**: Description of the ruleset.
- **source**: Source entity type for which the ruleset will apply.
- **target**: Target entity type for which the ruleset will apply.
- **status**: Status of the ruleset: Active or Inactive.

- **createdBy**: Determines who created the ruleset. System sets this value to the user ID who is logged on to the REST API, only when ruleset is created. The value provided by you will be ignored.
- **modifiedBy**: Determines who created the ruleset. System sets this value to the user ID who is logged on to the REST API, when ruleset is created or modified. The value provided by you will be ignored.
- **createdDate**: Determines when the ruleset was created. System sets this value to the current system date, when a ruleset is created. The value provided by you will be ignored.
- **modifiedDate**: Determines when the ruleset was created. System sets this value to the current system date, when a ruleset is created or modified. The value provided by you will be ignored.

A.1.5 RmTask Resource

Use this request structure in the API body when one of the request parameters is set as rmTaskResource.

The following are the parameters:

- **name** (mandatory): Name of the task to be registered or updated.
- **description**: Description of the task.
- **status**: Status of the task: Active or Inactive.

A.1.6 Entity Resource

Use this request structure in the API body when one of the request parameters is set as entityResource.

The following are the parameters:

- **id** (mandatory): ID of the entity. Entity type and id will be a unique combination in the entities table.
- **status**: Status of entity: Active or Inactive. This value is provided to activate or deactivate an entity. While creating a new entity, the value is not considered and always an Active entity is created. For updating an entity, use the PUT /resources/entities/{id} API. For information, see Section 4.6, "PUT /resources/entities/{id}". Status can be passed as request parameter (recommended) or in the body of the API.
- **type**: Details of the entity type.
  - **name**: Name of the entity type. It is mandatory while creating a new entity.
- **attributes**: An array of attributes for the entity.

  Each attribute contains the following fields:
  - **name** (mandatory): Attribute name of the attribute type.
  - **value**: Value for the attribute. This is mandatory for attribute type with mandatory flag as Y or indexed flag as Y. The format for the value depends on the data_type field of the attribute type. The following are the formats that can be used for the corresponding data types:
    - **String**: Any string value with the maximum length as specified in the size_value field of the attribute type.

  - **Char**: Any character value (string value with length of 1).
  - **Boolean**: Possible values are true and false.
  - **Integer**: Any integer value between max_value and min_value.
  - **Float**: Any float value between max_value and min_value.
  - **Double**: Any double value between max_value and min_value.
  - **Date**: Any value in the format *relationship.management.date.format* defined in the ohmpi_rm.properties file.
  - **Timestamp**: Any value in the format relationship.management.timestamp.format defined in the ohmpi_rm.properties file.
  - **Object**: Currently, not supported.

A.1.7 Entity Resources

Use this request structure in the API body when one of the request parameters is set as entityResources. This is mainly a collection of multiple entityResource.

The following are the parameters:

- **items**: An array storing multiple entityResource.
- **entityResource**: Contains details of an entity. For information, see Section A.1.6, "Entity Resource".

A.1.8 Relationship Resource

Use this request structure in the API body when one of the request parameters is set as relationshipResource.

The following are the parameters:

- **source**: Details of the source entity. For information, see Section A.1.6, "Entity Resource". ID and name of the entity type are mandatory parameters.
- **target**: Details of the target entity. For information, see Section A.1.6, "Entity Resource". ID and name of the entity type are mandatory parameters.
- **status**: Status of relationship: Active or Inactive. This value is provided to activate or deactivate a relationship. While creating a new relationship, the value is not considered and always an Active relationship is created. For updating a relationship, use the PUT /resources/relationships/{id} API. For information, see Section 5.1, "GET /resources/relationships/{id}". Status can be passed as request parameter (recommended) or in the body of the API.
- **validity**: Determines the validity of the relationship. Possible values are Potential and Resolved. If you do not provide any value, system sets the values as Resolved.
- **createdBy**: Determines who created the relationship. System sets this value to the user ID who is logged on to the REST API, when relationship is created. The value provided by you will be ignored. You cannot update the value later.
- **modifiedBy**: Determines who last modified the relationship. System sets this value to the user ID who is logged on to the REST API, whenever a relationship is created or modified. The value provided by you will be ignored.

- **createdDate**: Determines when the relationship was created. System sets this value to the current system date, only when a relationship is created. The value provided by you will be ignored. You cannot update the value later.
- **modifiedDate**: Determines when the relationship was last modified. System sets this value to the current system date, when a relationship is created or modified. The value provided by you will be ignored.
- **effectiveDate**: You can provide value for this and it must be passed in the date format *relationship.management.date.format* defined in the ohmpi_rm.properties file. If you do not provide a value, it is set to the current system date, when the relationship is created.
- **expirationDate**: Determines when the relationship expires. You can provide value for this and it must be passed in the date format *relationship.management.date.format* defined in the ohmpi_rm.properties file. If you do not provide a value, it is set to the current system date plus 10 years, from when the relationship is created.
- **attributes**: An array of attributes for the relationship.

  Each attribute contains the following fields:

  - **name** (mandatory): Attribute name of the attribute type.
  - **value**: Value for the attribute. This is mandatory for an attribute type with mandatory flag as Y or indexed flag as Y. The format for value depends on the data_type field of the attribute type. The following are the formats that can be used for corresponding data types:
    - **String**: Any string value with maximum length as specified in the size_value field of the attribute type.
    - **Char**: Any character value (string value with length of 1).
    - **Boolean**: Possible values are true and false.
    - **Integer**: Any integer value between max_value and min_value.
    - **Float**: Any float value between max_value and min_value.
    - **Double**: Any double value between max_value and min_value.
    - **Date**: Any value in the format *relationship.management.date.format* defined in the ohmpi_rm.properties file.
    - **Timestamp**: Any value in the format relationship.management.timestamp.format defined in the ohmpi_rm.properties file.
    - **Object**: Currently, not supported.

A.1.9 Integration Relationship Resource

Use this request structure in the API body when using integration APIs and one of the request parameters is set as relationshipResource.

The structure of the request remains the same as relationshipResource mentioned in Section A.1.8, "Relationship Resource", except that while creating a new relationship, you must also pass the following attributes in the attributes array:

- **LocalId**: Local ID of the entity that is created.
- **SystemCode**: System code in the MPI application to which the record belongs.

A.2 Response Structure

A.2.1 Entity Type Response Structure

The following structure is displayed in the response of an entity type REST API.

> **Note:** If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the entity type, use the GET /metadata/entities/{name} API with expand=all.

The following are the parameters:

- **entityTypeId**: System generated ID assigned to the entity type. It is always displayed.
- **name**: Name of the entity type. It is always displayed.
- **status**: Status of the entity type. It is always displayed.
- **description**: Description of the entity type. It is displayed when expand=all.
- **id**: Describes the details that are stored for the entity resource in this field. It is displayed when expand=all.
- **links**: For information, see Section A.3, "Links".
- **attributes**: Array of the attribute types for the entity type. It is displayed when expand=all.

  Each attribute type in the array contains the following details:

  - **attributeTypeId**: System generated ID assigned to the attribute type.
  - **name**: Name of the attribute type.
  - **description**: Description of the attribute type.
  - **type**: Data type of the attribute type.
  - **status**: Status of the attribute type.
  - **sizeValue**: Maximum size of data that can be stored for the attribute type.
  - **defaultValue**: Default value of the attribute type.
  - **minValue**: Minimum value of the attribute type.
  - **maxValue**: Maximum value of the attribute type.
  - **mandatory**: Mandatory flag of the attribute type.
  - **readOnly**: Read-only flag of the attribute type.
  - **indexed**: Indexed flag of the attribute type.
  - **links**: For information, see Section A.3, "Links".

A.2.2 Relationship Type Response Structure

The following structure is displayed in the response of a relationship type REST API.

> **Note:** If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the relationship type, use the GET /metadata/relationships/{name} API with expand=all.

The following are the parameters:

- **relationshipTypeId**: System generated ID assigned to the relationship type. It is always displayed.
- **name**: Name of the relationship type. It is always displayed.
- **status**: Status of the relationship type. It is always displayed.
- **description**: Description of the relationship type. It is displayed when expand=all.
- **start**: Details of the source entity type.
  - **name**: Name of the source entity type. It is always displayed.
  - **status**: Status of the source entity type. It is always displayed.
  - **links**: For information, see Section A.3, "Links". It is always displayed.
- **end**: Details of the target entity type.
  - **name**: Name of the target entity type. It is always displayed.
  - **status**: Status of the target entity type. It is always displayed.
  - **links**: For information, see Section A.3, "Links". It is always displayed.
- **direction**: Direction of the relationship type. It is displayed when expand=all.
- **multiplicity**: Multiplicity of the relationship type. It is displayed when expand=all.
- **createdBy**: Username of the user who created the relationship type. It is displayed when expand=all.
- **modifiedBy**: Username of the user who last modified the relationship type. It is displayed when expand=all.
- **createdDate**: Date when the relationship type was created. It is displayed when expand=all.
- **modifiedDate**: Date when the relationship type was last modified. It is displayed when expand=all.
- **effectiveDate**: Date from when the relationship type is effective. It is displayed when expand=all.
- **expirationDate**: Date when the relationship type expires. It is displayed when expand=all.
- **task**: ID of the task that is invoked when a new relationship of the relationship type is created. It is displayed when expand=all.
- **links**: For information, see Section A.3, "Links".
- **attributes**: Array of the attribute types for the relationship type. It is displayed when expand=all.

  Each attribute type in the array contains the following details:

  - **attributeTypeId**: System generated ID assigned to the attribute type.
  - **name**: Name of the attribute type.
  - **description**: Description of the attribute type.
  - **type**: Data type of the attribute type.
  - **status**: Status of the attribute type.
  - **sizeValue**: Maximum size of data that can be stored for the attribute type.
  - **defaultValue**: Default value of the attribute type.
  - **minValue**: Minimum value of the attribute type.
  - **maxValue**: Maximum value of the attribute type.
  - **mandatory**: Mandatory flag of the attribute type.
  - **readOnly**: Read-only flag of the attribute type.
  - **indexed**: Indexed flag of the attribute type.
  - **links**: For information, see Section A.3, "Links".

A.2.3 Domain Response Structure

The following structure is displayed in the response of a domain REST API.

> **Note:** If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the domain, use the GET /management/domains/{domain} API with expand=all.

The following are the parameters:

- **domainId**: System generated ID assigned to the domain. It is always displayed.
- **name**: Name of the domain. It is always displayed.
- **url**: URL of the MPI application for the domain. It is always displayed.
- **jndiname**: JNDI name of the MPI Application. It is always displayed.
- **userid**: Username of the user in WebLogic. It is always displayed.
- **usercode**: Password of the user. It is always displayed as *******.
- **objectXml**: Contents of the object.xml file. It is displayed when expand=all or expand=objectxml.
- **fields**: Fields in the object.xml file. Field name, field type, and field size are displayed. It is displayed when expand=all or expand=fields.
- **links**: For information, see Section A.3, "Links".

A.2.4 Ruleset Response Structure

The following structure is displayed in the response of a ruleset REST API.

> **Note:** If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the ruleset, use the GET /management/rulesets/{ruleset} API with expand=all.

The following are the parameters:

- **rulesetId**: System generated ID assigned to the ruleset. It is always displayed.
- **name**: Name of the ruleset. It is always displayed.
- **description**: Description of the ruleset. It is always displayed.
- **source**: Source entity type for which the ruleset will apply. It is always displayed.
- **target**: Target entity type for which the ruleset will apply. It is always displayed.
- **status**: Status of the ruleset. It is always displayed.
- **createdBy**: Username of the user who created the ruleset. It is always displayed.
- **modifiedBy**: Username of the user who created the ruleset. It is always displayed.
- **createdDate**: Date when the ruleset was created. It is always displayed.
- **modifiedDate**: Date when the ruleset was modified. It is always displayed.
- **ruleXml**: Contents of the xml file associated with the ruleset. It is displayed when expand=all or expand=rulexml.
- **links**: For information, see Section A.3, "Links".

A.2.5 Task Response Structure

The following structure is displayed in the response of a task REST API.

> **Note:** If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the task, use the GET /management/tasks/{task} API with expand=all.

The following are the parameters:

- **name**: Name of the task to be registered or updated.
- **description**: Description of the task.
- **status**: Status of the task: Active or Inactive.
- **script**: Contents of the groovy script file associated with the task. It is displayed when expand=all or expand=script.
- **links**: For information, see Section A.3, "Links".

A.2.6 Entity Response Structure

The following structure is displayed in the response of an entity REST API.

Note: If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed.

The following are the parameters:

- **entityId**: System generated ID assigned to the entity. It is always displayed.
- **id**: Unique iD provided by the user or received through MPI integration. It is always displayed.
- **status**: Status of the entity. It is always displayed.
- **links**: For information, see Section A.3, "Links".
- **type**: Details of the entity type associated with the entity. For information displayed in entity type, see Section A.2.1, "Entity Type Response Structure".
- **attributes**: Array of the attributes for the entity. It is displayed when expand=all.

  Each attribute in the array contains the following details:

  - **entityAttributeId**: System generated ID assigned to the attribute.
  - **name**: Name of the attribute.
  - **value**: Value assigned to the attribute.
  - **entity**: Details of the entity linked to this attribute.
    - **entityId**: System generated ID assigned to the entity.
    - **links**: For information, see Section A.3, "Links".

A.2.7 Relationship Response Structure

The following structure is displayed in the response of a relationship REST API.

Note: If the API has an expand option and the expand is not set to any value, only some of the attributes are displayed. If the expand is set to *all*, then all the values are displayed. APIs like POST or PUT might not have an expand option. For such APIs, to view the complete details of the relationship, use the GET /resources/relationships/{id} API with expand=all.

The following are the parameters:

- **relationshipId**: System generated ID assigned to the relationship. It is always displayed.
- **start**: Details of the source entity. For information on the details of entity that is displayed, see Section A.2.6, "Entity Response Structure".
- **end**: Details of the target entity type. For information on the details of entity that is displayed, see Section A.2.6, "Entity Response Structure".
- **type**: Details of the relationship type associated with the relationship. For information on the details displayed in relationship type, see Section A.2.7, "Relationship Response Structure".
- **status**: Status of the relationship. It is always displayed.

- **validity**: Validity of the relationship. It is always displayed.
- **createdBy**: Username of the user who created the relationship. It is displayed when expand=all.
- **modifiedBy**: Username of the user who last modified the relationship. It is displayed when expand=all.
- **createdDate**: Date when the relationship was created. It is displayed when expand=all.
- **modifiedDate**: Date when the relationship was last modified. It is displayed when expand=all.
- **effectiveDate**: Date from when the relationship is effective. It is displayed when expand=all.
- **expirationDate**: Date when the relationship expires. It is displayed when expand=all.
- **links**: For information, see Section A.3, "Links".
- **attributes**: Array of the attribute types for the relationship type. It is displayed when expand=all.
- **attributes**: Array of the attributes for the relationship. It is displayed when expand=all.

  Each attribute in the array contains the following details:

  - **relationshipAttributeId**: System generated ID assigned to the attribute.
  - **name**: Name of the attribute.
  - **value**: Value assigned to the attribute.
  - **relationship**: Details of the relationship linked to this attribute.
    - **relationshipId**: System generated ID assigned to the relationship.
    - **links**: For information, see Section A.3, "Links".

A.3 Links

The links resource displays information of the REST API invoked.

- **links**: Details of the API href, mediatype, and rel.
  - **href**: Link to access the resource which has the resource displayed.
  - **mediaType**: application/json
  - **rel**: canonical

A.4 Sample Files

The following sections provide the sample ruleset and task files used in some of the APIs.

A.4.1 Sample Ruleset File 1

```
<RmRules xmlns="http://www.w3.org"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
        xsi:schemaLocation="http://www.w3.org rules.xsd" >
    <RmRuleSet name="primary-care-physician-creation" source="IndProvider"
target="Patient" description="create relationship between IndProvider and
Patient">
    <rule name="primary-care-physician-creation" description="create relationship
between IndProvider and Patient">
        <if>
          <condition>
            <target name="Provider.ID" />
          </condition>
          <then>
            <task name="PrimaryCarePhysicianCreationTask"
arg="primary-care-physician-of"/>
          </then>
        </if>
      </rule>
    </RmRuleSet>
</RmRules>
```

A.4.2 Sample Ruleset File 2

```
<?xml version="1.0" encoding="UTF-8"?>
<RmRules xmlns="http://www.w3.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org rules.xsd" >
 <RmRuleSet name="CreatePatientOfSampleRuleset" source="Patient"
target="IndProvider" description="create relationship between the start to the
end" >
  <rule name="CreatePatientOfSampleRule" description="create relationship between
the start to the end">
    <if>
      <condition>
        <source name="FirstName" op="="  target="FirstName" />
      </condition>
      <then>
        <task name="CreatePatientOfSampleTask" arg="patient-of" arg1="Rule"/>
      </then>
    </if>
  </rule>
 </RmRuleSet>
</RmRules>
```

A.4.3 Sample Ruleset File 3

```
<?xml version="1.0" encoding="UTF-8"?>
<RmRules xmlns="http://www.w3.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org rules.xsd" >


  <RmRuleSet name="CreatePatientOfSampleRuleset"
source="Patient" target="IndProvider" description="create relationship between the
start to the end" >
    <rule name="CreatePatientOfSampleRule" description="create relationship
between the start to the end">
      <if>
        <condition>
```

```
          <source name="FirstName" op="="target="Names.FirstName" />
        </condition>
        <then>
          <task name="CreatePatientOfSampleTask" arg="patient-of" arg1="Rule"/>
        </then>
      </if>
    </rule>
  </RmRuleSet>
</RmRules>
```

A.4.4 Sample Task File 1

```
/*----------------------------------------------------------------*
 * Copyright (c) 2010 Oracle Corporation, Redwood Shores, CA, USA *
 *                  All rights reserved.                          *
 *---------------------------------------------------------------*/
package oracle.hsgbu.ohmpi.rm.rules.task;

import java.util.List;
import java.util.ArrayList;
import java.util.Date;
import oracle.hsgbu.ohmpi.rm.data.Entity;
import oracle.hsgbu.ohmpi.rm.data.EntityType;
import oracle.hsgbu.ohmpi.rm.data.KeyValue;
import oracle.hsgbu.ohmpi.rm.data.Relationship;
import oracle.hsgbu.ohmpi.rm.data.RelationshipType;
import oracle.hsgbu.ohmpi.rm.rules.service.RmTaskException;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskContext;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskInterface;
import oracle.hsgbu.ohmpi.rm.rules.service.data.DataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.data.RmDataObject;
import oracle.hsgbu.ohmpi.rm.service.RelationshipManagementService;
import oracle.hsgbu.ohmpi.rm.service.RelationshipManagementMetaData;

import oracle.hsgbu.ohmpi.rm.data.AttributeType;
import oracle.hsgbu.ohmpi.rm.data.EntityType;
import oracle.hsgbu.ohmpi.rm.data.RelationshipType;
import oracle.hsgbu.ohmpi.rm.data.KeyValue;
import oracle.hsgbu.ohmpi.rm.data.EntityAttribute;
import oracle.hsgbu.ohmpi.rm.data.Entity;
import oracle.hsgbu.ohmpi.rm.data.RelationshipAttribute;
import oracle.hsgbu.ohmpi.rm.data.Relationship;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskContext;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskInterface;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskMetaData;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskService;
import oracle.hsgbu.ohmpi.rm.rules.service.data.DataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.data.RmDataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.data.EntityDataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.data.LiteralDataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.RmTaskException;

/**
 * PrimaryCarePhysicianCreationTask in Groovy
 * @author ohmpi-team
 */
public class PrimaryCarePhysicianCreationTask implements
```

```
RmTaskInterface {

public PrimaryCarePhysicianCreationTask() {
}

public void execute(RmTaskContext context)
      throws RmTaskException {
    try {
    RmTaskMetaData rmMetadata = context.getRmTaskMetaData();
    RmTaskService rmService = context.getRmTaskService();
    EntityDataObject sourceData = context.getSourceData();
    EntityDataObject targetData = context.getTargetData();
    RmDataObject rmData = context.getRmData();
      if (sourceData != null &&
        rmData != null) {

        String relTypeName = rmData.getValue("arg");
        if (relTypeName.equalsIgnoreCase("primary-care-physician-of")) {
          if (sourceData.getName().equals("Patient")) {

            EntityType startType = rmMetadata.getEntityType("IndProvider");
            Entity indProviderEntity = new Entity();

indProviderEntity.setEId(sourceData.getValue("Provider.ID"));

        indProviderEntity.setType(startType);
          List<Entity> start =
rmService.searchEntity(indProviderEntity);//getEntityByEId(sourceData.getValue("Pr
ovider.ID"), startType);

          if (start != null) {
            EntityType endType = rmMetadata.getEntityType("Patient");
            Entity end = new Entity();
        end.setEId(sourceData.getValue("Id"))
            end.setType(endType);

            RelationshipType relType =
rmMetadata.getRelationshipType(relTypeName);
            List<KeyValue> attributes = new ArrayList<>();
            attributes.add(new KeyValue("createdBy", "byRule"));
            Relationship relation = rmService.createRelationship(start.get(0),
relType, end, attributes);
            List<Object> results = new ArrayList<>();
            results.add(relation);
            context.setResults(results);
          }
        } else {
            throw new RmTaskException("unsupported inbound domain '" +
sourceData.getName() + "'");
        }
      } else {
          throw new RmTaskException("unsupported relationship type '" +
relTypeName + "'");
      }
    } else {
```

```
                throw new RmTaskException("inbound message information
insufficient");
    }
   } catch (Throwable t) {
      throw new
RmTaskException("PrimaryCarePhysicianCreationTask failed to create a relationship"
+ t);
    }
  }
}
```

A.4.5 Sample Task File 2

```
/*-----------------------------------------------------------------*
 * Copyright (c) 2010 Oracle Corporation, Redwood Shores, CA, USA *
 *                 All rights reserved.                           *
 *-----------------------------------------------------------------*/
package oracle.hsgbu.ohmpi.rm.rules.task;


import java.util.List;
import java.util.ArrayList;
import java.util.Date;
import oracle.hsgbu.ohmpi.rm.data.Entity;
import oracle.hsgbu.ohmpi.rm.data.EntityType;
import oracle.hsgbu.ohmpi.rm.data.KeyValue;
import oracle.hsgbu.ohmpi.rm.data.Relationship;
import oracle.hsgbu.ohmpi.rm.data.RelationshipType;
import oracle.hsgbu.ohmpi.rm.rules.service.RmTaskException;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskContext;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskInterface;
import oracle.hsgbu.ohmpi.rm.rules.service.data.DataObject;
import oracle.hsgbu.ohmpi.rm.rules.service.data.RmDataObject;
import oracle.hsgbu.ohmpi.rm.service.RelationshipManagementService;
import oracle.hsgbu.ohmpi.rm.rules.task.RmTaskService;
import oracle.hsgbu.ohmpi.rm.rules.service.data.EntityDataObject;


/**
 * CreatePatientOfSampleTask in Groovy
 * @author ohmpi-team
 */
public class CreatePatientOfSampleTask implements RmTaskInterface {


  public CreatePatientOfSampleTask() {
  }


  public void execute(RmTaskContext context) throws RmTaskException {
    try {
      RmTaskService rmService = context.getRmTaskService();
      RmTaskMetaData rmMetadata = context.getRmTaskMetaData();
      EntityDataObject sourceData = context.getSourceData();
      EntityDataObject targetData = context.getTargetData();
      RmDataObject rmData = context.getRmData();


      if (sourceData != null && targetData != null && rmData != null) {
        EntityType startType = rmMetadata.getEntityType("Patient");
        EntityType endType = rmMetadata.getEntityType("IndProvider");
        Entity start = new Entity();
        start.setEId(sourceData.getValue("Id"));
        start.setType(startType);
        Entity end = new Entity();
        end.setEId(targetData.getValue("Id"));
        end.setType(endType);
        RelationshipType relType =
rmMetadata.getRelationshipType(rmData.getValue("arg"));
        String arg1 = rmData.getValue("arg1");
        List<KeyValue> attributes = new ArrayList<>();
        attributes.add(new KeyValue("createdBy", arg1));
        Relationship relation = rmService.createRelationship(start, relType, end,
attributes);
        List<Object> results = new ArrayList<Object>();
        results.add(relation);
        context.setResults(results);
      }
    } catch (Throwable t) {
        throw new RmTaskException("CreatePatientOfSampleTask failed to create a
relationship" + t);
    }
  }
}
```