

Oracle® Healthcare Precision Medicine

Administrator's Guide

Release 1.1

E76044-02

April 2017

Oracle Healthcare Precision Medicine Administrator's Guide, Release 1.1

E76044-02

Copyright © 2016, 2017 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	v
Documentation Accessibility	v
Finding More Information	v
Related Documents	v
1 EMR Configuration	
1.1 Overview	1-1
1.2 Configuration	1-1
1.3 Configuring the OPSS Keystore.....	1-3
1.3.1 Using the Enterprise Manager	1-3
1.3.2 Using the WebLogic Scripting Tool	1-6
1.3.3 Granting Permission to Use the Keystore	1-6
2 External System Configurations	
2.1 Dalliance.....	2-1
2.1.1 Dalliance Configuration.....	2-1
2.1.1.1 Retrieving a List of Genome Configurations.....	2-1
2.1.1.2 Adding a Genome Configuration	2-2
2.1.2 Dalliance Configuration by Genome Version	2-3
2.1.2.1 Retrieving Configuration for a Genome Version.....	2-3
2.1.2.2 Updating an Existing Genome Configuration	2-3
2.1.2.3 Deleting a Genome Version Configuration	2-4
2.2 N-Of-One.....	2-4
2.2.1 N-of-One Credential Configuration.....	2-4
2.2.1.1 Getting Credentials Configured for N-of-One	2-5
2.2.1.2 Adding a Credential for Invoking N-of-One.....	2-5
2.2.1.3 Updating Credentials for Invoking N-of-One.....	2-6
2.2.1.4 Deleting Credential Used for Invoking N-of-One	2-6
2.3 Enabling or Disabling Features Related to External Systems.....	2-7
2.3.1 Returning State of Current Features	2-7
2.3.2 Enabling or Disabling Features in Oracle Healthcare Precision Medicine	2-7
3 Published Report Template Configuration	
3.1 Configuring the Logo in the Report Header and Footer.....	3-1
3.2 Configuring the Text in the Report Header and Footer	3-1

4 Annotation Pane Configuration

4.1	GET Annotation Pane Configuration.....	4-1
4.2	UPDATE Annotation Pane Configuration.....	4-2
4.3	UPDATE Annotation Pane Labels.....	4-3
4.3.1	Request Body for Different Element Types	4-4
4.4	UPDATE Annotation Pane Display	4-5
4.4.1	Request Body for Different Element Types	4-6
4.5	REORDER Annotation Pane Elements	4-7
4.5.1	Request Body for Different Element Types	4-7

5 Loader API

5.1	Fetch Job Status and Log.....	5-1
5.1.1	Job Status.....	5-1
5.1.2	Job Log.....	5-1
5.2	Data Ingestion with Loaders	5-2
5.2.1	Preferred Transcript Loader.....	5-2

6 Report Statistics APIs

6.1	Retrieving Aggregate Report Count	6-1
A.1	Additional Configuration Parameters	A-1

Index

Preface

This guide describes how to perform administration tasks in Oracle Healthcare Precision Medicine (OHPM).

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 that have purchased support 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.

Finding More Information

Oracle Help Center

The latest user documentation for Oracle Health Sciences products is available at <http://docs.oracle.com/en/industries/health-sciences/>.

My Oracle Support

The latest release notes, patches and white papers are on My Oracle Support (MOS) at <https://support.oracle.com>. For help with using MOS, see https://docs.oracle.com/cd/E74665_01/MOSHP/toc.htm.

Related Documents

Oracle Business Intelligence Enterprise Edition Documentation

The OBIEE documentation is available at <http://docs.oracle.com/middleware/12212/biee/index.html>.

Oracle Healthcare Foundation

The OHF documentation is available at <https://docs.oracle.com/en/industries/health-sciences/oracle-healthcare-foundation/index.html>

EMR Configuration

This section describes EMR configuration. It contains the following topics:

- [Section 1.1, "Overview"](#)
- [Section 1.2, "Configuration"](#)
- [Section 1.3, "Configuring the OPSS Keystore"](#)

1.1 Overview

To publish genomic reports to EMR, you must configure OHPM to integrate with EPIC EMR. This section describes how to configure EMR.

Multiple EMR integration configurations can be created and maintained in OHPM. However, at any given point in time, only one configuration will remain in the active status. If not, the genomic report will not be published to EMR from OHPM.

1.2 Configuration

Perform the following steps to configure OHPM for EPIC EMR:

Note: Steps 1 to 4 in the following list should be performed only if EMR configuration was *not* done during installation.

1. Note down the host name and the TLS port number where EPIC EMR listens for incoming lab result message.

This TLS port typically is the EPIC Interconnect HL7 V2 interface that can handle base64 encoded attachments. Oracle strongly recommends that you use TLS 1.2 as it is secure.
2. Get the Certificate Authority (CA) Certificate of the CA that signed the digital certificates used by EPIC TLS.

CA certificates are generally from standard public CAs like Verisign. If EPIC was configured to use a self-signed certificate, then use the local CA certificate employed to sign the certificate request.
3. If you have completed EMR configuration through the installer, proceed to step 5. If not, proceed to step 3.
4. Load the CA Certificate into OHPM WebLogic Server's Oracle Platform Security Services (OPSS) keystore. You can do this either:
 - [Using the Enterprise Manager](#)

- [Using the WebLogic Scripting Tool](#)
5. Grant permission to read the keystore.
 6. Use the EMR Integration Configuration REST service available as part of the OHPM application to configure details required to publish a report to the EMR using the TLS keystore created in the previous step.

The service end point of the EMR Integration Configuration REST service and definition of its payload are as follows:

Table 1–1 Service End Point

HTTP Method	URL	Description
GET/DELETE	http://<host>:<port>/ohpm/opmemrintegration/resources/opm/api/v1.0/emrconfigservice/emrconfig/{emrId}	Retrieve or Delete EMR integration configuration for a given EMR ID
GET	http://<host>:<port>/ohpm/opmemrintegration/resources/opm/api/v1.0/emrconfigservice/emrconfig/status/{status}	Retrieve EMR integration configuration for a given status. Status is active <i>A</i> or Inactive <i>I</i> .
POST/PUT	http://<host>:<port>/ohpm/opmemrintegration/resources/opm/api/v1.0/emrconfigservice/emrconfig	Create or Update using the EMRIntegrationConfig payload

Table 1–2 Definition of Payload

Field (XML/JSON)	Description	Valid Value Set	Example
emrIntegrationConfig	Root Element	-	-
createdById	Created by user ID (automatically populated)	-	-
createdOnDt	Date when record was created (automatically populated)	-	-
description	Description of the EMR	User defined	EPIC EMR
emrId	A unique ID for the configuration. Used to update.	User defined	Epic_1
emrType	Type of the EMR	EPIC	EPIC
integrationHost	Host name or IP of the remote EMR system for TCP/TLS communication, otherwise NULL	Remote host name/IP	remotehost.com
integrationPort	Port number for TCP/TLS communication, otherwise NULL	Valid remote host port number	8088
integrationUrl	SOAP/REST URL for SOAP/REST end point, otherwise NULL	Valid URL	https://host:7002/epic/fhir/svc
integrationAuthMethod	Authentication method for calling remote service	HTTP-BASIC, WS-SECURITY	HTTP-BASIC
integrationAuthUsername	Username for HTTP-BASIC and WS-SECURITY. Credentials must be available in OPSS/JPS configuration	User defined	testuser
integrationSslKeystore	OPSS/JPS keystore name in stripe <i>OPMApp</i> . Mandatory when the integration URL starts with https.	User defined	castore
messageFormat	The format of the EMR message to be generated	HL7V2	HL7V2
messageTemplateLoc	Location of the template. The default location for EPIC is templates/EPIC_HL7V2.template	Valid file location	templates/EPIC_HL7V2.template
receivingApplication	The name of the application receiving the HL7V2 message.	User defined	EPIC
receivingFacility	The name of the facility receiving the HL7V2 message.	User defined	EPIC-LAB

Table 1–2 (Cont.) Definition of Payload

Field (XML/JSON)	Description	Valid Value Set	Example
reportTestName	Test name (code) to represent genomic test result, based on what EMR accepts	User defined	GENE^GENOMIC EXAM
rowWid	Internally generated surrogate primary key	-	-
sendingApplication	The name of the application sending the HL7V2 message.	User defined	ORACLE-MI
sendingFacility	The name of the facility sending the HL7V2 message.	User defined	GENETIC-LAB
status	Status of the configuration record	A (active), I (inactive)	A
transportProtocol	Transport protocol supported by this EMR	TCP, TLS, SOAP1, SOAP2, REST	TLS

Following is a sample XML payload for the REST service to create a TLS connection configuration to EPIC:

```
<?xml version="1.0" encoding="UTF-8"?>
<emrIntegrationConfig>
  <description>EPIC EMR Configuration</description>
  <emrId>Epic_Id1</emrId>
  <emrType>EPIC</emrType>
  <integrationHost>localhost</integrationHost>
  <integrationPort>8081</integrationPort>
  <integrationSslKeystore>emrtruststore</integrationSslKeystore>
  <messageFormat>HL7V2</messageFormat>
  <messageTemplateLoc>templates/EPIC_HL7V2.template</messageTemplateLoc>
  <receivingApplication>EPIC</receivingApplication>
  <receivingFacility>EPC</receivingFacility>
  <reportTestName>GENOMIC^REPORT</reportTestName>
  <sendingApplication>ORACLE-PM</sendingApplication>
  <sendingFacility>ORACLE-PM</sendingFacility>
  <status>A</status>
  <transportProtocol>TLS</transportProtocol>
</emrIntegrationConfig>
```

7. Enable publishing the report to EMR. For details, see [Section 2.3, "Enabling or Disabling Features Related to External Systems"](#).

1.3 Configuring the OPSS Keystore

You can configure the OPSS keystore using either the WebLogic Scripting Tool or the Enterprise Manager. For standard setups, follow the WLST steps from [Section 1.3.2](#). If you have an Enterprise Manager license, follow the steps from [Section 1.3.1](#).

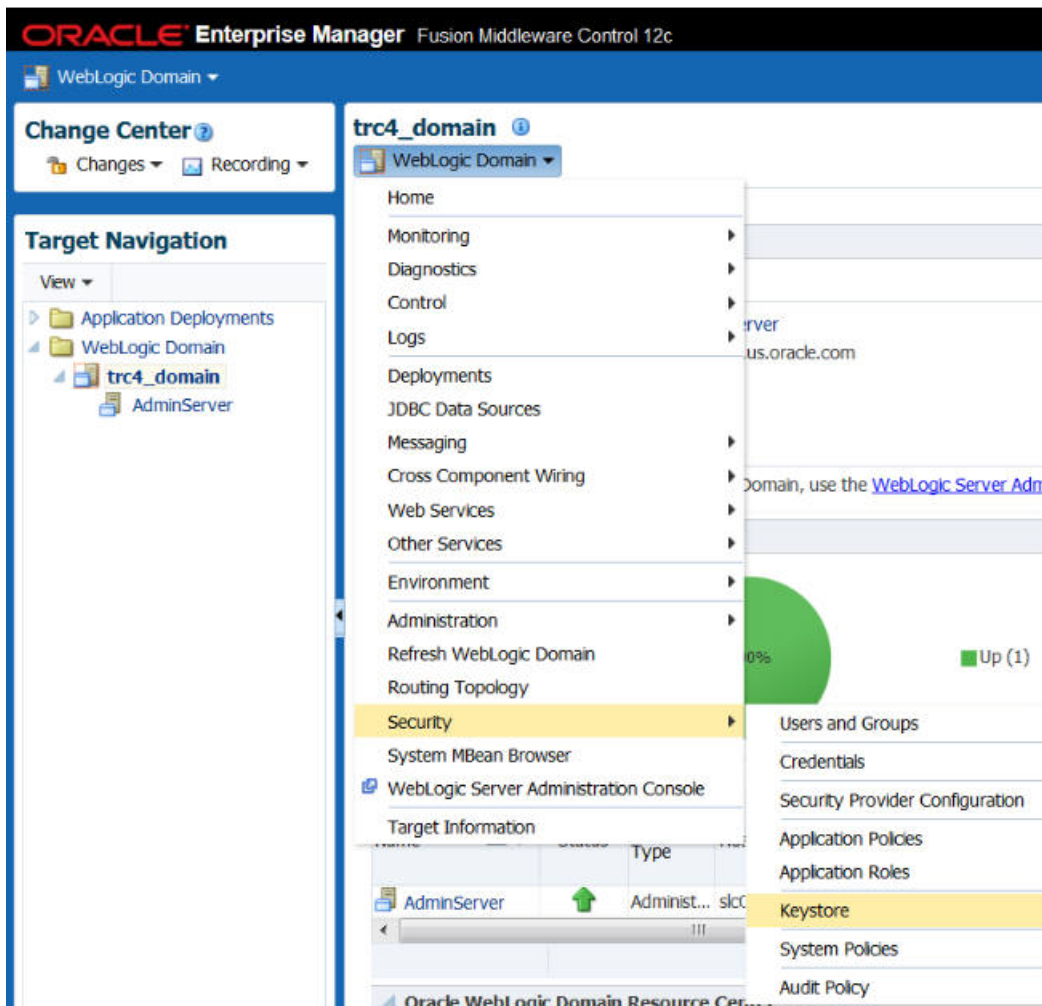
1.3.1 Using the Enterprise Manager

Note: The Enterprise Manager requires a separate license and is not part of the OHPM installer components.

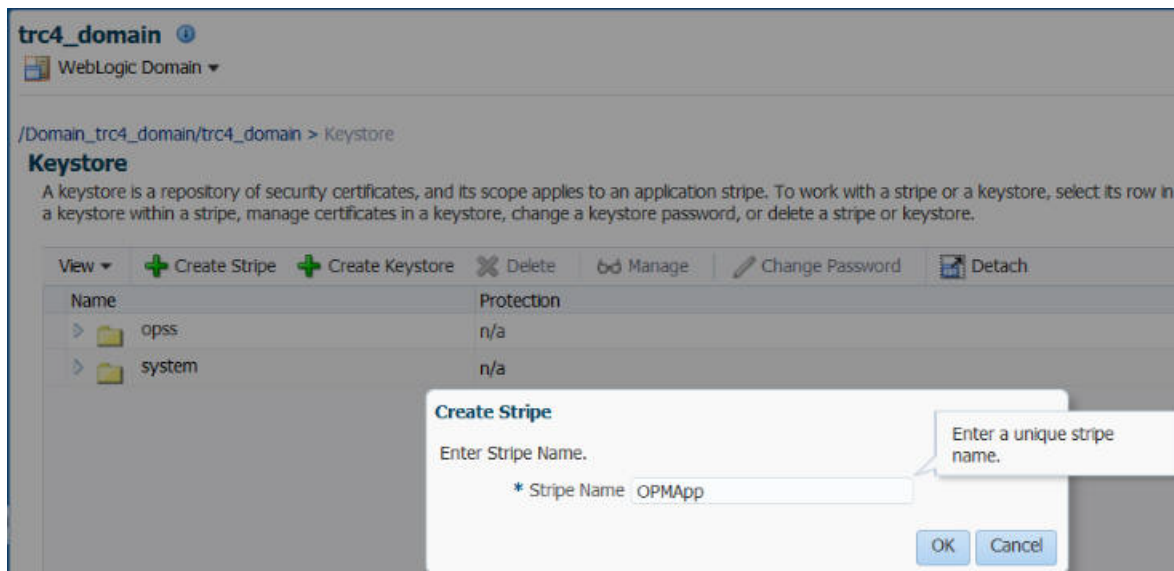
To configure the EMR keystore after using the OHPM installer, use the Enterprise Manager. This involves the following steps:

1. Log into WebLogic Enterprise Manager Fusion Middleware Control.

- Navigate to **WebLogic Domain > Security > Keystore**.



- Click **Create Stripe** to create a new stripe named OPMApp.



4. Click **OK**.
5. Click **Create Keystore**.
6. Create a new keystore that will be used in the EMR integration configuration. For example, emrtruststore.

Create Keystore

Keystore Stripe Name: OPMApp

* Keystore Name: emrtruststore

Protection: Policy Password

Keystore Password:

Confirm Password:

Grant Permission:

Code Base URL:

OK Cancel

7. Ensure that **Policy** is selected for **Protection**.
8. Deselect **Grant Permission**.
9. Click **OK**.

For details on creating a keystore, see the section on *Creating a Keystore with Fusion Middleware Control in the Oracle® Fusion Middleware Securing Applications with Oracle Platform Security Services* available at the following location

<https://docs.oracle.com/middleware/1213/idm/app-security/kssadm.htm#CACHHCH>

10. Select the keystore created in the previous step and click **Manage**.

trc4_domain WebLogic Domain

Information
The keystore, OPMApp/emrtruststore, has been created.

/Domain_trc4_domain/trc4_domain > Keystore

Keystore
A keystore is a repository of security certificates, and its scope applies to an application stripe. To work with a stripe or a keystore, a keystore within a stripe, manage certificates in a keystore, change a keystore password, or delete a stripe or keystore.

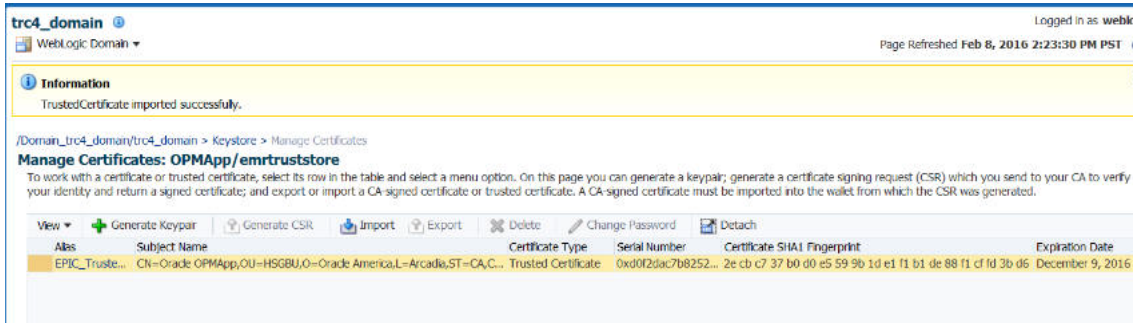
Name	Protection
opss	n/a
OPMApp	n/a
emrtruststore	Policy
system	n/a

View + Create Stripe + Create Keystore ✗ Delete Manage Change Password Detach

Manage certificates in a keystore

11. Click **Import**.
12. Select the **Certificate Type** as **Trusted Certificate**.

13. Enter an **Alias**.
14. You can either browse and select the CA certificate file or copy the certificate content into the text field provided.
15. Click **OK**. The CA trusted certificate is successfully added to the keystore.



For details on creating a keystore, see the section on Importing a Certificate or Trusted Certificate with Fusion Middleware Control in the *Oracle® Fusion Middleware Securing Applications with Oracle Platform Security Services* available at the following location

<https://docs.oracle.com/middleware/1213/idm/app-security/kssadm.htm#CACDEAJH>

1.3.2 Using the WebLogic Scripting Tool

Perform the following steps to configure the OPSS keystore using WLST.

1. Load the CA Certificate into Oracle Platform Security Services (OPSS) using the WebLogic Scripting Tool (WLST). Use the following commands:

```
$ cd $MIDDLEWARE_HOME/wlserver/common/bin
$ ./wlst.sh
wls:/offline> connect()
Please enter your username: '<weblogic admin user>'
Please enter your password: '<weblogic admin user password>'
Please enter your server URL [t3://localhost:7001]: '<weblogic admin url>'
wls:/ohpm_domain/serverConfig> svc = getOpssService(name='KeyStoreService')
wls:/ohpm_domain/serverConfig> svc.createKeyStore(appStripe='OHF-Opm-App',
name='emrtruststore', password='', permission=true)
wls:/ohpm_domain/serverConfig>
svc.importKeyStoreCertificate(appStripe='OPMApp', name='emrtruststore',
password='', alias='EPIC_Trusted_Cert', keypassword='',
type='TrustedCertificate', filepath='<full path to the CA certificate file>')
```

1.3.3 Granting Permission to Use the Keystore

Perform the following steps to grant permission to use the keystore created in OPSS using WLST.

1. Connect to the WebLogic admin server using wlst.
2. Perform the steps in [Section 1.3.2](#).
3. Execute the following WLST command:

```
wls:/offline> connect('<weblogic admin user>', '<weblogic admin user
password>', '<weblogic admin url ex: t3://localhost:7001>')
wls:/ohpm_domain/serverConfig> svc = getOpssService(name='KeyStoreService')
```

```
wls:/ohpm_domain/serverConfig> grantPermission(appStripe="OHF-Opm-App",
codeBaseURL='file:${oracle.deployed.app.dir}/OHF-Opm-App${oracle.deployed.app.e
xt}',
permClass="oracle.security.jps.service.keystore.KeyStoreAccessPermission",
permTarget="stripeName=OPMApp,keystoreName=emrtruststore,alias=*",
permActions="read")
```

External System Configurations

This chapter details configuration APIs for external systems used in OHPM. These are REST based services and are authenticated using BASIC authentication. Admin user (having pm_admin_group role) can use these services for configuration.

It includes the following topics:

- [Section 2.1, "Dalliance"](#)
- [Section 2.2, "N-Of-One"](#)
- [Section 2.3, "Enabling or Disabling Features Related to External Systems"](#)

Common Request Header

All API invocations with base path as /ohpm/opmconfigapi/ should have header value **X-Requested-By** set with the value OPM. For more information on configuring a sample REST client to access the OHPM API REST based services, see Article ID 2148026.1 on My Oracle Support.

2.1 Dalliance

REST API's to manage genome version details required by the Dalliance genome browser. DAS server details are required for configuring Dalliance.

2.1.1 Dalliance Configuration

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/dalliance
ACCEPTS	application/json
RETURNS	application/json

2.1.1.1 Retrieving a List of Genome Configurations

Method

GET - Genome configurations

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
-----------	-------------	----------------	------------------------------

Response

HTTP Status: 200 (success)

Body: Example

```
{
  "name": "HG19",
  "authority": "GRCh",
  "version": "38",
  "sequenceTrackUrl": "<URL>",
  "genesTrackUrl": "<URL>",
  "dnaVersion": "V68"
},
{
  "name": "HG18",
  "authority": "NCBI",
  "version": "37",
  "sequenceTrackUrl": "<URL>",
  "genesTrackUrl": "<URL>",
  "dnaVersion": "V68"
}
```

2.1.1.2 Adding a Genome Configuration

Method

POST - Add a genome configuration. You will have to provide DAS server URLs that are used in Dalliance to plot variant details.

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre>{ "name": "<UCSC_NAME>", "authority": "<authority_name>", "version": "<version_number>", "sequenceTrackUrl": "<DAS server sequence track URL>", "genesTrackUrl": "<DAS server sequence track URL>" "dnaVersion": "<DNA Reference Build Version>" }</pre>

Response

Code	Reason	Representation
200	success	<pre>{ "success": "true" }</pre>

2.1.2 Dalliance Configuration by Genome Version

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/dalliance/{genomeVersion}
ACCEPTS	application/json
RETURNS	application/json

2.1.2.1 Retrieving Configuration for a Genome Version

Method

GET - Configuration for a genome version

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
genomeVersion	Genome version	path	String

Response

Code	Reason	Representation (Example)
200	success	<pre>{ "name": "HG19", "authority": "GRCh", "version": "38", "sequenceTrackUrl": "<URL>", "genesTrackUrl": "<URL>" "dnaVersion": "V68" }</pre>

2.1.2.2 Updating an Existing Genome Configuration

Method

PUT - Update an existing genome configuration

Request

PARAMETER	DESCRIPTIO N	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
genomeVersio n	Genome version	path	string
body		body	<pre>{ "name": "<UCSC_NAME>", "authority": "<authority_name>", "version": "<version_number>", "sequenceTrackUrl": "<URL>", "genesTrackUrl": "<URL>" "dnaVersion": "<DNA Reference Build Version>" }</pre>

Response

Code	Reason	Representation
200	success	<pre>{ "success": "true" }</pre>

2.1.2.3 Deleting a Genome Version Configuration**Method**

DELETE - Configuration for a genome version

Request

PARAMETER	DESCRIPTIO N	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
genomeVersio n	Genome version	path	string

Response

Code	Reason	Representation (Example)
200	success	

2.2 N-Of-One

The following REST APIs have been provided to manage configurations for invoking N-of-One genomic interpretation APIs. Enable N-of-One related features after configuring N-of-One details. For details, see [Section 2.3, "Enabling or Disabling Features Related to External Systems"](#).

2.2.1 N-of-One Credential Configuration

PATH	
	<server url>/ohpm/opmconfigapi/v1.0/config/credential/nof1

ACCEPTS	application/json
RETURNS	application/json

2.2.1.1 Getting Credentials Configured for N-of-One

Method

GET - User credential for N-of-One

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
-----------	-------------	----------------	------------------------------

Response

Code	Reason	Representation
200	success	<pre>{ "id": "1002", "credentialType": "ORG", "authorizationToken": null, "productKey": "<productKey>", "customerId": "<customerId>" }</pre>

authorizationToken is not loaded in GET.

2.2.1.2 Adding a Credential for Invoking N-of-One

Method

POST - Add credential details for invoking N-of-One

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre>{ "authorizationToken": "<AUTH TOKEN>", "productKey": "<productKey>", "customerId": "<customerId>" }</pre>

Response

Code	Reason	Representation
200	success	<pre>{ "id": "1231", "credentialType": "ORG", "authorizationToken": "<AUTH TOKEN>", "productKey": "<productKey>", "customerId": "<customerId>" }</pre>

2.2.1.3 Updating Credentials for Invoking N-of-One**Method**

PUT - Update credential details for N-of-One

Request

PARAMETER	DESCRIPTIO N	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre>{ "authorizationToken": "<AUTH TOKEN>", "productKey": "<productKey>", "customerId": "<customerId>" }</pre>

Response

Code	Reason	Representation
200	success	<pre>{ "authorizationToken": "<AUTH TOKEN>" "productKey": "<productKey>", "customerId": "<customerId>" }</pre>

2.2.1.4 Deleting Credential Used for Invoking N-of-One**Method**

DELETE - Delete credential used for invoking N-of-One

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
-----------	-------------	----------------	------------------------------

Response

Code	Reason	Representation
200	success	<pre>{ "success": "true" }</pre>

2.3 Enabling or Disabling Features Related to External Systems

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/externalFeaturesState
ACCEPTS	application/json
RETURNS	application/json

2.3.1 Returning State of Current Features**Method**

GET - Returns current features state

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
-----------	-------------	----------------	------------------------------

Response

Code	Reason	Representation
200	success	<pre>{ "dalliance":false, "clinvar":true, "nofl":false, "emr":false }</pre>

2.3.2 Enabling or Disabling Features in Oracle Healthcare Precision Medicine**Method**

PUT - Enable or disable features in OHPM

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre>{ "dalliance":false, "clinvar":true, "nofl":false, "emr":true }</pre>

Response

Code	Reason	Representation
200	success	<pre>{ "success": "true" }</pre>

Published Report Template Configuration

This section describes how an admin user can configure the text and logo that appear in header and footer of the published PDF report. It includes the following sections:

- [Section 3.1, "Configuring the Logo in the Report Header and Footer"](#)
- [Section 3.2, "Configuring the Text in the Report Header and Footer"](#)

3.1 Configuring the Logo in the Report Header and Footer

The report logo should be stored on domain_home on the server and the paths (including the logo name) should be configured using admin APIs.

Images for the logo should be of 0.75" × 0.75" dimension (height*width) so that the logo is not stretched or squeezed.

Use the following API to configure the report logo:

Method	PUT
API Path	<server url>/ohpm/opmconfigapi/v1.0/config/{configurationName}
JSON Body	:{ "Key": "value " }
Example	host:port/ohpm/opmconfigapi/v1.0/config/PUBLISH_REPORT_HEADER_LOGO1 { "PUBLISH_REPORT_HEADER_LOGO1": " ./DynImages/Footer.jpg " }

Configuration Key Names for header and footer logo images are:

- Header Logo
 - PUBLISH_REPORT_HEADER_LOGO1
 - PUBLISH_REPORT_HEADER_LOGO2
- Footer Logo
 - PUBLISH_REPORT_FOOTER_LOGO

3.2 Configuring the Text in the Report Header and Footer

Admin user can configure the text in the report header and footer using the same API that is used to configure the logo.

Configuration Names for header and footer text are:

- Report Header: Text at top of report header, 3 lines
 - PUBLISH_REPORT_HEADER_LINE1
 - PUBLISH_REPORT_HEADER_LINE2
 - PUBLISH_REPORT_HEADER_LINE3
- Report Footer: Text at bottom of report, 2 lines
 - PUBLISH_REPORT_FOOTER_LINE1
 - PUBLISH_REPORT_FOOTER_LINE2

Annotation Pane Configuration

This section describes how to configure the annotation pane using REST APIs. It contains the following sections:

- [Section 4.1, "GET Annotation Pane Configuration"](#)
- [Section 4.2, "UPDATE Annotation Pane Configuration"](#)
- [Section 4.3, "UPDATE Annotation Pane Labels"](#)
- [Section 4.4, "UPDATE Annotation Pane Display"](#)
- [Section 4.5, "REORDER Annotation Pane Elements"](#)

4.1 GET Annotation Pane Configuration

Gets all the annotation pane UI details as JSON.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/annotationPane
ACCEPTS	application/json
RETURNS	application/json
METHOD	GET

Response

Code	Reason	Representation (Example)
200	success	

Example

```
{
  "eltWid": 1,
  "type": "PANE",
  "level": 0,
  "displayOrder": 0,
  "label": "Annotation",
  "items": [{
    "eltWid": 2,
    "type": "INPUT",
    "level": 1,
    "displayOrder": 0,
    "label": "Significance",
    "data_type": "ENUM",
    "attr_wid": 57,
    "data_key": "significance",
```

```

    "options": {
      "multiple": false,
      "read_only": false,
      "display": true,
      "allow_custom": false
    },
    "values": [{
      "wid": 2,
      "label": "Likely Benign",
      "value": 2,
      "displayOrder": 3
    }, {
      "wid": 1,
      "label": "Unknown Significance",
      "value": 1,
      "displayOrder": 4
    }
  ]
}, {
  "eltWid": 3,
  "type": "INPUT",
  "level": 1,
  "displayOrder": 1,
  "label": "Associated Diagnosis",
  "data_type": "LOOKUP",
  "attr_wid": 71,
  "data_key": "var_associated_diseases",
  "data_url": "diagnoses",
  "options": {
    "multiple": true,
    "read_only": false,
    "display": true,
    "allow_custom": true
  }
}
}
}

```

4.2 UPDATE Annotation Pane Configuration

This updates the label, display (True or False) and Display Order on the Annotation pane. The user has to be admin role of pm_admin_group.

Note: While updating, make sure the display order is correct. This value cannot be duplicated for all siblings.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/annotationPane/changed
ACCEPTS	application/json
RETURNS	application/json
METHOD	PUT

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	[["eltWid":Number, "label":String, "display" : String, "displayOrder" : String]]

Example

```
[[
  {
    "eltWid": 2,
    "label": "Significance",
    "display": true,
    "displayOrder": 0
  }, {
    "eltWid": 3,
    "label": "Associated Diagnosis",
    "display": true,
    "displayOrder": 1
  }
]]
```

Response

Code	Reason	Representation (Example)
200	success	

4.3 UPDATE Annotation Pane Labels

This lets the user update the Label element on the annotation pane. The user must have the admin role of pm_admin_group.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/panes/annotation/labels
ACCEPTS	application/json
RETURNS	application/json
METHOD	PUT

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	{ "eltType": String, "eltDataKey": String, "label": String }

Example

```
[[
  {
    "eltType": "SECTION",
    "eltDataKey": "external_information",
  }
]]
```

```

    "label": "Externals"
  },
  {
    "eltType": "INPUT",
    "eltDataKey": "significance",
    "label": "Variant Significance"
  }
]

```

Response

Code	Reason	Representation (Example)
200	success	

4.3.1 Request Body for Different Element Types

The following table shows examples of request body for different element types.

Element Types	Body
Group SECTION Label	<pre> [{ "eltType": "SECTION", "eltDataKey": "external_information", "label": "Externals" }] </pre>
INPUT Label	<pre> [{ "eltType": "INPUT", "eltDataKey": "significance", "label": "Variant Significance" }] </pre>
SECTION Label for COMBO	<pre> [{ "eltType": "SECTION", "eltDataKey": "clinvar", "label": "ClinVar Info" }] </pre>
INPUT Label for COMBO: set it null	<pre> [{ "eltType": "INPUT", "eltDataKey": "clinvar", "label": null }] </pre>
INPUT Label for COMBO: set it value	<pre> [{ "eltType": "INPUT", "eltDataKey": "n_of_one", "label": "Send to Interpret" }] </pre>
INPUT Label: set it value	<pre> [{ "eltType": "INPUT", "eltDataKey": "drug_response_phenotype", "label": "Drug Response" }] </pre>
Error: Wrong eltType	<pre> [{ "eltType": "PANEL", "eltDataKey": null, "label": "Annotation Pane" }] </pre>
Error: Wrong eltDataKey	<pre> [{ "eltType": "INPUT", "eltDataKey": "significance1", "label": "Variant Significance" }] </pre>

Element Types	Body
Error: Element is marked as deleted eltDataKey	<pre>[{ "eltType": "INPUT", "eltDataKey": "drug_response_phenotype", "label": "Variant Drug Response Phenotype" }]</pre>

4.4 UPDATE Annotation Pane Display

This lets the user manipulate the display of SECTION/INPUT elements (hide or show) on the annotation pane. The user must have the admin role of pm_admin_group.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/panes/annotation/display
ACCEPTS	application/json
RETURNS	application/json
METHOD	PUT

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre>[{ "eltDataKey": String, "position": String, "refPointEltDataKey": String }]</pre> <p>Note: Elements referenced by eltDataKey and refPointEltDataKey must be on the same level on the Annotation Pane layout.</p>

Example

```
[{
  "eltDataKey": "cell_line",
  "position": "first",
  "refPointEltDataKey": null
},
{
  "eltDataKey": "clinvar",
  "position": "last",
  "refPointEltDataKey": null
},
{
  "eltDataKey": "external_information",
  "position": "after",
  "refPointEltDataKey": "cell_line"
},
{
  "eltDataKey": "significance",
  "position": "before",
  "refPointEltDataKey": "comments"
}]
```

Response

Code	Reason	Representation (Example)
200	success	

4.4.1 Request Body for Different Element Types

The following table shows examples of request body for different element types.

Element Types	Body
Group SECTION Hide For group section Show or display=true, an error is generated. You do not need to show any empty group section. But if you need to show at least one child INPUT or COMBO INPUT (SECTION + INPUT) - this group section is automatically displayed.	<pre>[{ "eltType": "SECTION", "eltDataKey": "external_information", "display": "false" }]</pre>
SECTION for COMBO Hide/Show	<pre>[{ "eltType": "SECTION", "eltDataKey": "clinvar", "display": "false" }]</pre>
INPUT for COMBO Hide/Show	<pre>[{ "eltType": "INPUT", "eltDataKey": "clinvar", "label": "ClinVar Info" }]</pre>
INPUT Label for COMBO: set it null	<pre>[{ "eltType": "INPUT", "eltDataKey": "clinvar", "label": null }]</pre>
INPUT	<pre>[{ "eltType": "INPUT", "eltDataKey": "significance", "display": "false" }]</pre>
Error: Wrong eltType	<pre>{ "eltType": "SECTION", "eltDataKey": "external_information", "display": "true" }</pre> <pre>{ "errorMessage": "Unsupported action for UI element SECTION was requested", "httpMessage": "Internal Server Error", "httpStatusCode": 500, "appErrorCode": "19-02787-47284", "errorMessageKey": "MI_CONFIG_ELEMENT_TYPE_UNSUPPORTED_ACTION" }</pre>
Error: Wrong eltDataKey	<pre>[{ "eltType": "INPUT", "eltDataKey": "significancel", "display": "false" }]</pre> <pre>{ "errorMessage": "Querying ui element table produced no results for annotation", "httpMessage": "Internal Server Error", "httpStatusCode": 500, "appErrorCode": "19-02784-95113", "errorMessageKey": "MI_CONFIG_EMPTY_ANNOTATION_QUERY_RESULT" }</pre>

Element Types	Body
Error: Element is marked as deleted eltDataKey	<pre> [{ "eltType": "INPUT", "eltDataKey": "drug_response_phenotype", "display": "false" }] </pre>
<pre> { "errorMessage": "Querying ui element table produced no results for annotation", "httpMessage": "Internal Server Error", "httpStatusCode": 500, "appErrorCode": "19-02783-48347", "errorMessageKey": "MI_CONFIG_EMPTY_ ANNOTATION_QUERY_RESULT" } </pre>	

4.5 REORDER Annotation Pane Elements

This lets the user reorder SECTION/INPUT elements on the same level of the annotation pane. The user must have the admin role of pm_admin_group.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/panes/annotation/reorder
ACCEPTS	application/json
RETURNS	application/json
METHOD	PUT

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	<pre> [{ "eltType": String, "eltDataKey": String, "display": String }] </pre>

Example

```

[ {
  "eltType": "SECTION",
  "eltDataKey": "external_information",
  "display": "false"
},
{
  "eltType": "SECTION",
  "eltDataKey": "clinvar",
  "display": "false"
} ]

```

Response

Code	Reason	Representation (Example)
200	success	

4.5.1 Request Body for Different Element Types

The following table shows examples of request body for different element types.

Element Types	Body
Move external_information after cell_line on the Annotation Pane: No change – default order	<pre>[{ "eltDataKey": "external_information", "position": "after", "refPointEltDataKey": "cell_line" }]</pre>
Move external_information as first element on the Annotation Pane	<pre>[{ "eltDataKey": "external_information", "position": "first", "refPointEltDataKey": null }]</pre>
Move external_information as last element on the Annotation Pane	<pre>[{ "eltDataKey": "external_information", "position": "last", "refPointEltDataKey": null }]</pre>
Restore external_information after cell_line on the Annotation Pane	<pre>[{ "eltDataKey": "external_information", "position": "after", "refPointEltDataKey": "cell_line" }]</pre>
Restore n_of_one after clinvar in the External Information Section	<pre>[{ "eltDataKey": " n_of_one", "position": "after", "refPointEltDataKey": "clinvar" }]</pre>

This section describes the APIs for OHPM loaders. It contains the following topics:

- [Section 5.1, "Fetch Job Status and Log"](#)
- [Section 5.2, "Data Ingestion with Loaders"](#)

5.1 Fetch Job Status and Log

You can view the latest job status (running or completed), progress and the completion status (success or failure). If the job failed, you can see more information on the error by referring the job logs.

For Authorization, provide a user that is in the allowed role (pm_admin_group)

5.1.1 Job Status

This provides the current status for the Loader Job.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/loader/jobstatus/{jobId}
ACCEPTS	application/json
RETURNS	application/json
METHOD	GET

Response

Code	Reason	Representation (Example)
200	success	<pre>{ "status": "COMPLETE", "links": [{ "rel": "logs", "href": "/ohpm/opmconfigapi/v1.0/config/loader/jobstatus/{jobId}/logs" }] }</pre>
500	error	error message object

5.1.2 Job Log

This provides detailed log information for the job.

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/loader/jobstatus/{jobId}/logs
ACCEPTS	application/json
RETURNS	application/json
METHOD	GET

Response

Code	Reason	Representation (Example)
200	success	[{ "insertDate": string, "logDetail": string, "errorCode": string, "errorInfo": string }]
500	error	error message object

5.2 Data Ingestion with Loaders

5.2.1 Preferred Transcript Loader

Loads the preferred transcript data files into the OHPM environment.

Allowed Roles for API: pm_admin_group

Loader Procedure: load_pref_transcr_stg

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/loader/preferred-transcript
ACCEPTS	application/json
RETURNS	application/json
METHOD	POST

Request

PARAMETER	DESCRIPTION	PARAMETER TYPE	DATA TYPE/ REPRESENTATION
body		body	{ "file": "{bucketName}/{objectName}", "readSize":Number, "fullRefresh" : boolean }

- *readSize* is optional in the request
- *fullRefresh* is optional and if not specified, the default value is false.

Note: For the *file* input parameter, specify the *bucketName* and *objectName* that was used in the File Upload Service when uploading the file. If you have manually uploaded the file, then directly specify the file name.

Example

```
{
  "file": "mypreferred/preferred_transcript_2020"
  "readSize" : 8
}
```

Response

Code	Reason	Representation (Example)
200	success	HTTP/1.1 202 Accepted Location: /ohpm/opmconfigapi/v1.0/config/loader/jobstatus/{jobId} Note: The Location points to the URI for the JobId. Use this to fetch the job status.
500	error	error message object

Report Statistics APIs

This chapter describes REST API's that can be invoked by administrators to get the statistics for tracking and pricing. It contains the following topics:

6.1 Retrieving Aggregate Report Count

This REST API retrieves aggregates on report generation statistics based on different input parameters.

PATH	<server url>/ohpm/opmconfigapi/v1.0/stats/admin/report/count
ACCEPTS	application/json
RETURNS	application/json
METHOD	GET

The following input parameters are optional.

If any of the parameters are not provided, the search is not limited. For example, if the status is null, then both draft and published reports are included.

Parameter	Description	Parameter Type	Data Type (Representation)
status	Report Status (Draft or Published)	Query Parameter	String
user	List of users that created the report. If there is more than one user use a comma to separate the names.	Query Parameter	String
startdate	Date from which the search is started. Use the format YYYY/MM/DD.	Query Parameter	String
enddate	Date till which the search is performed. Use the format YYYY/MM/DD.	Query Parameter	String
includeDeleted	Indicates if deleted reports should be included. The default value is true.	Query Parameter	String

Request Headers

Content-Type: application/json

X-Requested-By: OPM

Example Request

HTTP GET:

http://localhost:7101/ohpm/opmapiconfig/v1.0/stats/admin/report/count?status=DRAFT

Response

Code	Reason	Response Type	Representation (Example)
200	success		Number (count)
500	error	error message object	

Miscellaneous Configuration

This section lists additional application configuration parameters that can be modified and the service that performs this.

A.1 Additional Configuration Parameters

Parameter Name	Default/Current Value	Description
DIAGNOSIS.DEFAULT_PROXIMITY_DAYS	5	Configuration to specify a date range, based on the order date, to retrieve diagnoses. For example if the order date is 10 Dec 2016, then fetch all the diagnosis within order date 5 Dec 2016 and 15 Dec 2016
DEFAULT_DATE_FORMAT	MM-dd-yyyy	Default date format
VARIANT_FILTERS_RESULTS_PAGE_SIZE	15	Configuration used to display numbers of rows in variant filtering table.
TEXTAREA_ROWS_DEFAULT	5	Defines numbers of rows in a small text area
TEXTAREA_ROWS_LARGE	10	Defines numbers of rows in large text area.

To configure the parameters in the above table, use the following API:

PATH	<server url>/ohpm/opmconfigapi/v1.0/config/{configurationName}
METHOD	PUT
JSON BODY	{ "Key": "value " }
EXAMPLE URL	host:port/ohpm/opmconfigapi/v1.0/config/TEXTAREA_ROWS_DEFAULT
EXAMPLE RESPONSE	{ "TEXTAREA_ROWS_DEFAULT": "10" }

C

configuration

dalliance, 2-1

emr, 1-1

n-of-one, 2-4

configure annotation pane

GET, 4-1

UPDATE, 4-2, 4-3, 4-5, 4-7

configure published report

logo, 3-1

text, 3-1

D

data ingestion with loaders, 5-2

E

external system features

disabling, 2-7

enabling, 2-7

J

job log, 5-1

job status, 5-1

L

loader design, 5-1

P

preferred transcript loader, 5-2

