

Oracle® Fusion Middleware

Using Oracle API Manager

12c (12.1.3)

E58404-06

November 2016

Introduces Oracle API Manager and describes how to install and administer the product and how to curate, discover, and use APIs.

Oracle Fusion Middleware Using Oracle API Manager, 12c (12.1.3)

E58404-06

Copyright © 2015, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

Contributing Authors:

Contributors:

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	vii
Audience	vii
Documentation Accessibility	vii
Conventions.....	vii
Part I Introduction to Oracle API Manager	
1 Introducing Oracle API Manager	
1.1 What is Oracle API Manager?.....	1-1
1.2 Understanding Oracle API Manager Features	1-2
1.3 Understanding User Roles in Oracle API Manager.....	1-2
1.4 Getting Started with Oracle API Manager	1-3
Part II Installing and Configuring Oracle API Manager	
2 Installing Oracle API Manager	
2.1 Planning the Installation.....	2-1
2.1.1 Product Distribution	2-1
2.1.2 Verifying Certification, System Requirements, and Interoperability	2-1
2.2 Installing Software Required for an API Manager Installation	2-2
2.2.1 Installing a Certified JDK	2-2
2.2.2 Installing and Configuring a Certified Database.....	2-3
2.2.3 Installing Oracle Fusion Middleware 12c Infrastructure.....	2-3
2.2.4 Installing the Oracle Service Bus Software	2-4
2.3 Installing Oracle API Manager via OPatch.....	2-4
2.4 Installing Required Patches.....	2-5
2.4.1 Installing the RCU Patch	2-5
2.5 Next Steps	2-5
3 Configuring the Oracle API Manager Domain	
3.1 Creating the Database Schemas.....	3-1
3.1.1 Installing and Configuring a Certified Database.....	3-1

3.1.2	Starting the Repository Creation Utility (RCU).....	3-1
3.1.3	Navigating the RCU Screens to Create the Schemas	3-2
3.2	Configuring Your WebLogic Domain	3-5
3.2.1	Starting the Configuration Wizard	3-5
3.2.2	Navigating the Configuration Wizard Screens to Configure the Domain.....	3-5
3.3	Starting the Servers.....	3-15
3.3.1	Starting the Node Manager.....	3-16
3.3.2	Starting the Administration Server.....	3-16
3.3.3	Starting the Managed Servers.....	3-17
3.4	Next Steps	3-17

4 Managing Users in Oracle API Manager

4.1	Managing Users	4-1
4.1.1	Create Groups for API Manager Roles.....	4-1
4.1.2	Assign Application Roles to Groups Using Fusion Middleware Control	4-2
4.1.3	Creating API Manager Users.....	4-4
4.1.4	Add Groups to the Monitors Parent Group	4-5
4.1.5	Assigning Users to Groups	4-6
4.2	Next Steps	4-7

Part III Using Oracle API Manager

5 Creating Your APIs

5.1	Accessing the Oracle Service Bus Console	5-1
5.2	Creating Proxy Services to Be Published as APIs	5-1
5.3	Activating the Session	5-2
5.4	Next Steps	5-3

6 Curating and Publishing Your APIs Using Oracle Service Bus

6.1	Editing the Metadata of an API	6-1
6.2	Understanding API Status.....	6-3
6.3	Making an API Private.....	6-3
6.4	Publishing an API to the Oracle API Manager Portal	6-4
6.4.1	Deciding if an API Should Be Managed or Unmanaged.....	6-4
6.4.2	Changing the Status of an API to Published	6-4
6.5	Hiding an API from the API Manager Portal	6-5
6.6	Deprecating an API	6-5
6.7	Next Steps	6-6

7 Using the Oracle API Manager Portal

7.1	Introducing the Oracle API Manager Portal.....	7-1
7.2	Accessing the API Manager Portal.....	7-1
7.2.1	Signing in to the API Manager Portal	7-1

7.2.2	Logging Out of the API Manager Portal.....	7-2
7.3	Managing Applications.....	7-2
7.3.1	Creating an Application	7-2
7.3.2	Subscribing to APIs from the Subscriptions Page	7-3
7.3.3	Editing an Application Description	7-4
7.3.4	Removing Applications from the API Manager Portal	7-4
7.4	Discovering APIs	7-4
7.4.1	Finding APIs Using the Catalog Page	7-5
7.4.2	Viewing Details of an API.....	7-5
7.4.3	Data Displayed on API Detail Pages	7-6
7.4.4	Subscribing to APIs from the Catalog Page.....	7-7
7.5	Viewing Your Subscriptions	7-8
7.5.1	Viewing Your Subscribed APIs.....	7-9
7.5.2	Viewing APIs Subscribed to a Specific Application.....	7-9
7.6	Using APIs in Your Applications	7-9
7.6.1	Understanding the Difference Between Managed and Unmanaged APIs.....	7-9
7.6.2	Using Unmanaged APIs.....	7-10
7.6.3	Using Managed APIs	7-10
7.6.4	Finding an Application's Entitlement Key	7-10
7.6.5	Testing APIs	7-11
7.7	Unsubscribing from APIs	7-11
7.7.1	Unsubscribe from APIs Using the APIs Panel of the Subscriptions Page	7-11
7.7.2	Unsubscribe from APIs Using the Applications Panel of the Subscriptions Page ...	7-12
7.8	Next Steps	7-12

8 Administering Oracle API Manager

8.1	Viewing Subscription Statistics Using the Subscriptions Page.....	8-1
8.1.1	Viewing a User's Subscriptions	8-1
8.1.2	Viewing API Subscription Details	8-1
8.1.3	Using the Quick Search Features	8-2
8.2	Viewing Analytics in Oracle API Manager.....	8-2
8.2.1	Enabling API Analytics	8-2
8.2.2	Viewing Simple Analytics for an API.....	8-3
8.2.3	Viewing Detailed Analytics for an API.....	8-4
8.2.4	Analytics Data Displayed on API Detail Pages	8-5
8.3	Importing and Exporting APIs and API Curation Metadata	8-6
8.3.1	Exporting from an Environment.....	8-6
8.3.2	Sample Script for Exporting Curation Metadata	8-7
8.3.3	Importing into an Environment	8-7
8.3.4	Sample Script for Importing Curation Metadata.....	8-8

Preface

Oracle Fusion Middleware User's Guide for Oracle API Manager introduces Oracle API Manager and describes how to install and administer the product and how to curate, discover, and use APIs.

Audience

This document is intended for curators, consumers, and administrators of APIs in Oracle API Manager, developers who create proxy services in Oracle Service Bus, and anyone interested in learning about the product.

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.

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.
<i>italic</i>	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.

Part I

Introduction to Oracle API Manager

This part introduces you to Oracle API Manager.

This part includes the following chapters:

- [Introducing Oracle API Manager](#)

Introducing Oracle API Manager

This chapter introduces Oracle API Manager, describes its features, and provides next steps to get started.

The following topics are covered:

- [What is Oracle API Manager?](#)
- [Understanding Oracle API Manager Features](#)
- [Understanding User Roles in Oracle API Manager](#)
- [Getting Started with Oracle API Manager](#)

1.1 What is Oracle API Manager?

Applications increasingly use Application Programming interfaces (APIs). These APIs can be developed by internal or external entities to access enterprise systems of record. The number of APIs available increases daily. Their usage is multiplied by a growing population of mobile devices and computing platforms (Web, tablets, smartphones, TVs, cars, and other household devices--the Internet of Things). As the number of APIs an organization produces and uses increases, the management and visibility of these APIs becomes increasingly important.

Oracle API Manager facilitates the creation of APIs that expose functionality of backend systems or other services. These APIs are published for use by application developers and are managed and monitored at runtime.

API Manager is installed with a standalone Oracle Service Bus instance, with which developers can create WSDL-based SOAP proxy services and REST proxy services to be published as APIs.

Note:

Proxy services configured with deprecated WLS 9 policies can not be published as APIs.

API Curators decide which of these services should be managed and exposed to developers. API Curators then add metadata and publish the APIs to the API Manager Portal, from which application developers subscribe to and view information necessary to consume the APIs. Administrative users can use the API Manager Portal to monitor usage metrics and performance analytics at runtime.

[Understanding Oracle API Manager Features](#), describes API Manager use cases and the API management lifecycle.

1.2 Understanding Oracle API Manager Features

With API Manager, you can create an API, publish it to a portal, and search for APIs of interest. You can also create applications and subscribe to APIs.

- **Create an API:** Use Service Bus facilities to create proxy services to publish as APIs for a variety of backend services.

See [Creating Your APIs](#) and [Creating and Configuring Proxy Services in *Developing Services with Oracle Service Bus*](#) for more information about developing proxy services.

- **Add descriptive metadata and publish API to Portal:** Users with the API Curator role add decorative metadata to APIs, decide if APIs should be managed, and publish APIs to the API Manager Portal.

See [Curating and Publishing Your APIs Using Oracle Service Bus](#) for more information about API curation tasks.

- **Discover and understand APIs:** Users with the API Consumer role use the API Manager Portal to search for APIs of interest, explore their metadata and test the API to understand its capabilities.

See [Discovering APIs](#) for more information about discovering and subscribing to APIs.

- **Create applications and subscribe to APIs:** Applications that reflect your organization's applications must be created in API Manager. After applications are created, users with the API Consumer role can subscribe APIs to applications.

See [Managing Applications](#) for more information about managing applications and [Subscribing to APIs from the Subscriptions Page](#) for information about subscribing to APIs.

- **Access the APIs at runtime:** After users have subscribed to APIs they are interested in, users can use an endpoint URL and entitlement key to access the backend service (published as an API).

See [Using APIs in Your Applications](#) for information about consuming APIs.

- **Use analytics to understand API performance:** Users with the API Admin role monitor subscription data and various runtime analytics with the API Manager Portal.

See [Viewing Subscription Statistics Using the Subscriptions Page](#) and [Viewing Analytics in Oracle API Manager](#) for more information about viewing API usage statistics and performance analytics.

The user roles discussed in this section are described in [Understanding User Roles in Oracle API Manager](#).

1.3 Understanding User Roles in Oracle API Manager

Certain roles are available for use with API Manager, and some Service Bus roles are also used with API Manager.

The following roles have been created for use with API Manager:

- **API Curator:** Users with this role use the Service Bus Console to curate the metadata associated with and publish proxy services as APIs to the API Manager

Portal. Users assigned this role have read-only access to the remainder of the Service Bus Console so they can view artifact information.

- **API Consumer:** Users with this role can use the API Manager Portal to view, subscribe to, and consume APIs. Users can also view runtime metrics for managed APIs. Users assigned this role have no access to the Service Bus Console.
- **API Admin:** Users with this role have administrative privileges for the API Manager Portal. API Admin users can view API usage metrics, analytics information, and detailed subscription information for each User and API. Users assigned this role have no access to the Service Bus Console.

The following Service Bus roles are also used with API Manager:

- **Developer:** Users with this role use the Service Bus Console to create proxy services that can be published as APIs on the API Manager Portal.
- **Administrator:** Users with this role have full access to the Service Bus Console and the API Manager Portal, in addition to full access to Oracle WebLogic Server. Users assigned this role cannot subscribe to assets using the API Manager Portal.
- **Deployer:** Users assigned this role are responsible for deploying the APIs, including both Service Bus artifacts and curation information. This role maps to the existing Deployer Service Bus role, with the additional responsibility of importing/exporting curation metadata.
- **Monitor:** Users assigned this role are responsible for monitoring

For more information about the existing user roles, see "Defining Access Security for Oracle Service Bus" in *Administering Oracle Service Bus*.

Creating users and assigning roles to these users in API Manager is described in [Managing Users in Oracle API Manager](#)

1.4 Getting Started with Oracle API Manager

After you have installed and configured API Manager, you curate APIs with Service Bus, discover and use APIs in the Portal, and various other administrative tasks.

Refer to the topics in [Installing and Configuring Oracle API Manager](#) to install and configure API Manager. This part also describes creating users.

After completing the tasks described in [Installing and Configuring Oracle API Manager](#), see [Using Oracle API Manager](#) for chapters on curating APIs using Oracle Service Bus, Discovering and Using APIs with the API Manager Portal, and Administration features in API Manager.

Part II

Installing and Configuring Oracle API Manager

This section details how to install Oracle API Manager and how to and deploy API Manager to a WebLogic Server instance.

This section contains the following chapters:

- [Installing Oracle API Manager](#)
- [Configuring the Oracle API Manager Domain](#)
- [Managing Users in Oracle API Manager](#)

Installing Oracle API Manager

This chapter describes how to install the Oracle API Manager software.

The following topics are covered:

- [Planning the Installation](#)
- [Installing Software Required for an API Manager Installation](#)
- [Installing Oracle API Manager via OPatch](#)
- [Installing Required Patches](#)
- [Next Steps](#)

2.1 Planning the Installation

This section describes topics you must be familiar with prior to installing Oracle API Manager, including where to acquire the software, verifying that your system meets the operating requirements, and installing software that is required for an API Manager installation.

2.1.1 Product Distribution

You must obtain the installation files from the Oracle Software Delivery Cloud located at

<https://edelivery.oracle.com/>

Note:

You must accept the license agreement before you download the installation patch.

2.1.2 Verifying Certification, System Requirements, and Interoperability

Oracle recommends that you use the certification matrix and system requirements documents in conjunction with each other to verify that your environment meets the necessary requirements for installation.

- [Verifying Your Environment Meets Certification Requirements](#)
- [Using the System Requirements Document to Verify Certification](#)
- [Verifying Interoperability Among Multiple Products](#)

2.1.2.1 Verifying Your Environment Meets Certification Requirements

Make sure that you are installing your product on a supported hardware and software configuration. For more information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

Oracle has tested and verified the performance of your product on all certified systems and environments; whenever new certifications occur, they are added to the proper certification document right away. New certifications can occur at any time, and for this reason the certification documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

2.1.2.2 Using the System Requirements Document to Verify Certification

Oracle recommends that you use the *Oracle Fusion Middleware System Requirements and Specifications* document to verify that the requirements of the certification are met. For example, if the certification document indicates that your product is certified for installation on 64-Bit Oracle Linux 5, this document should be used to verify that your Oracle Linux 5 system has met the required minimum specifications, like disk space, available memory, specific platform packages and patches, and other operating system-specific items. System requirements can be updated at any time, and for this reason the system requirement documents are kept outside of the documentation libraries and are available on Oracle Technology Network.

2.1.2.3 Verifying Interoperability Among Multiple Products

The *Understanding Interoperability and Compatibility* document defines interoperability, defines compatibility, and describes how multiple Fusion Middleware products from the same release or mixed releases may be used with each other. You should read this document if you are planning to install multiple Fusion Middleware products on your system.

2.2 Installing Software Required for an API Manager Installation

You must complete the tasks described in this section to install the required software before installing API Manager. You must install a certified JDK, install and configure a certified database, install the Oracle Fusion Middleware 12c infrastructure, and install Oracle Service Bus 12.1.3.

This section describes the following topics:

- [Installing a Certified JDK](#)
- [Installing and Configuring a Certified Database](#)
- [Installing Oracle Fusion Middleware 12c Infrastructure](#)
- [Installing the Oracle Service Bus Software](#)

2.2.1 Installing a Certified JDK

Before you can install Oracle API Manager, you must download and install a supported JDK on your system before attempting to run the installer.

Make sure that the JDK is installed outside of the Oracle home. The Oracle Universal Installer validates that the designated Oracle home directory is empty, and the install will not progress until an empty directory is specified. If you install JDK under Oracle home, you will experience issues in future operations. So, it is recommended that you

use the `/home/oracle/products/jdk` directory for the location of your JDK installation. You can then use the `java -jar` command to run the installer JAR file.

At the time this document was published, Oracle Fusion Middleware products required JDK 7.0 Update 55 or later. However, you should always verify the required JDK version by reviewing the certification information, as described in [Verifying Certification, System Requirements, and Interoperability](#).

To download the required JDK, use your browser to navigate to the following URL and download the Java SE JDK: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

2.2.2 Installing and Configuring a Certified Database

Oracle API Manager requires database schemas prior to domain configuration. If you do not already have a database where you can install these schemas, you must install and configure a certified database.

To find a certified database for your operating system, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

To make sure your database is properly configured for schema creation, see "Verifying Requirements for Oracle Repository Creation Utility" in the *Oracle Fusion Middleware System Requirements and Specifications* document.

Note:

For 12c, only a certified Oracle database can be used for the Oracle Fusion Middleware Infrastructure standard installation topology.

See the following references for more information about creating your product schemas:

- The Repository Creation Utility (RCU) is the tool used to create schemas in your database. This tool is available by installing the Oracle Fusion Middleware Infrastructure. For more information, see *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
- See [Creating the Database Schemas](#) for the specific schemas required by Oracle API Manager along with instructions to create the schemas.
- Refer to *Creating Schemas with the Repository Creation Utility* for more information about the Repository Creation Utility.

Note:

After installing a certified database and installing the product software, you must install the RCU patch as described in [Installing the RCU Patch](#) before you can create the product schemas.

2.2.3 Installing Oracle Fusion Middleware 12c Infrastructure

After you have installed the JDK and installed and configured a certified database, you must install an Oracle WebLogic Server instance. After installing Oracle API

Manager and creating the required database schemas, you configure a domain and deploy the application to a WebLogic server instance.

The latest version of Oracle WebLogic Server is available by installing the Oracle Fusion Middleware Infrastructure.

Note:

You must install Oracle WebLogic server using the Oracle Fusion Middleware 12c (12.1.3.0.0) Infrastructure installer (Part Number V44416-01) that includes Java Required Files (JRF). JRF consists of components not typically included in WebLogic Server installations that provide common functionality for Oracle business applications and application frameworks.

You must obtain this installer from the Oracle Fusion Middleware 12c Media Pack on the Oracle Software Delivery Cloud located at

<https://edelivery.oracle.com/>

Installing WebLogic Server using another installer will cause the API Manager installation to fail.

For more information about installing and configuring the Oracle Fusion Middleware Infrastructure, see *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

2.2.4 Installing the Oracle Service Bus Software

Before installing API Manager, you must install an instance of Oracle Service Bus 12.1.3 into the same Oracle home into which you installed the Oracle Fusion Middleware infrastructure. See "Planning Your Oracle Service Bus Installation" and "Installing the Oracle Service Bus Software" in *Installing and Configuring Oracle Service Bus* for more information about installing Service Bus.

Note:

You do not need to deploy Oracle Service Bus to a WebLogic Server domain. The Service Bus domain template will be selected and deployed as a part of the API Manager domain configuration, as described in [Configuring the Oracle API Manager Domain](#).

Extending an existing Service Bus domain with an API Manager domain is not supported in this release.

2.3 Installing Oracle API Manager via OPatch

Install API Manager on top of an Oracle Service Bus 12.1.3 installation by acquiring the product installation zip file from Oracle Software Delivery Cloud and applying the installation patch using OPatch. See *Patching with OPatch* for more information about using OPatch.

To install API Manager:

1. Acquire the API Manager installation zip file from the Oracle Software Delivery Cloud at <https://edelivery.oracle.com/>.

2. Extract the contents of the installation zip file to a local directory. Verify that the extracted directory contains the installation patch:
`p20225320_121300_Generic.zip`
3. Follow the instructions in the patch's `README.txt` file to complete the installation.

2.4 Installing Required Patches

Before you create the database schemas and configure the WebLogic domain, you must install the required RCU patch by completing the following task:

- [Installing the RCU Patch](#)

2.4.1 Installing the RCU Patch

Install the required RCU patch by using the `OPatch` utility. This makes required changes to the SOA Suite database schemas when running the Repository Creation Utility before deploying an API Manager domain. See *Patching with OPatch* for more information about using `OPatch`.

To install the required RCU patch:

1. Ensure that you have acquired the API Manager installation zip file from the Oracle Software Delivery Cloud and extracted its contents to a local directory, as described in [Installing Oracle API Manager via OPatch](#).
2. Verify that the extracted directory contains the RCU patch:
`p20311552_121300_Generic.zip`.
3. Follow the instructions in the patch's `README.txt` file.

2.5 Next Steps

The next steps include creating the required database schemas and configuring the Oracle WebLogic Server domain for API Manager.

See [Configuring the Oracle API Manager Domain](#) to complete the necessary tasks.

Configuring the Oracle API Manager Domain

This chapter describes how to create and configure the WebLogic Server domain for Oracle API Manager.

By the end of this chapter, you would have created database schemas for Oracle API Manager and configured an Oracle WebLogic domain with one cluster and two managed servers on a single host.

The following topics are covered:

- [Creating the Database Schemas](#)
- [Configuring Your WebLogic Domain](#)
- [Starting the Servers](#)
- [Next Steps](#)

3.1 Creating the Database Schemas

Before you can configure an Oracle API Manager domain, you must install the required schemas on a database certified for use with this release of Oracle API Manager.

Follow the instructions in this section to install these schemas:

- [Installing and Configuring a Certified Database](#)
- [Starting the Repository Creation Utility \(RCU\)](#)
- [Navigating the RCU Screens to Create the Schemas](#)

3.1.1 Installing and Configuring a Certified Database

Make sure you have installed and configured a certified database, and that the database is up and running.

For more information, see "Installing a Database and Database Schemas" in *Planning an Installation of Oracle Fusion Middleware*.

3.1.2 Starting the Repository Creation Utility (RCU)

Note:

Before creating product schemas, ensure that you have already applied the RCU patch as described in [Installing the RCU Patch](#).

To start the Repository Creation Utility (RCU):

1. Navigate to the `ORACLE_HOME/oracle_common/bin` directory on your system.

2. Start RCU:

On UNIX operating systems:

```
./rcu
```

On Microsoft Windows operating systems:

```
rcu.bat
```

3.1.3 Navigating the RCU Screens to Create the Schemas

Follow the instructions in this section to create the schema for Oracle API Manager:

- [Introducing RCU](#)
- [Selecting a Method of Schema Creation](#)
- [Providing Database Credentials](#)
- [Specifying a Custom Prefix and Selecting Schemas](#)
- [Specifying Schema Passwords](#)
- [Specifying Custom Variables](#)
- [Completing Schema Creation](#)

3.1.3.1 Introducing RCU

Click **Next**.

3.1.3.2 Selecting a Method of Schema Creation

If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.

If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option will generate an SQL script, which can be provided to your database administrator. See "Understanding System Load and Product Load" in *Creating Schemas with the Repository Creation Utility*.

3.1.3.3 Providing Database Credentials

Provide the database connection details for RCU to connect to your database. You must provide the following information:

- Database Type
- Host Name
- Port
- Service Name
- Username

- Password
- Role

Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

3.1.3.4 Specifying a Custom Prefix and Selecting Schemas

Select **Create new prefix**, specify a custom prefix, then select the **SOA Suite** schema. This will automatically select SOA Infrastructure, along with the following schemas as dependencies:

- Metadata Services
- Audit Services
- Audit Services Append
- Audit Services Viewer
- Oracle Platform Security Services
- User Messaging Service
- WebLogic Services

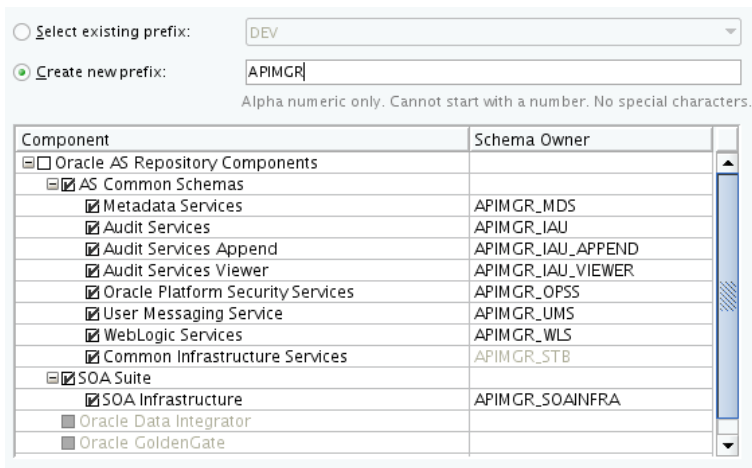
A schema called Common Infrastructure Services is also automatically created; this schema is grayed out and cannot be selected or deselected. This schema enables you to retrieve information from RCU during domain configuration. For more information, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

Tip:

For more information about custom prefixes, see "Understanding Custom Prefixes" in *Creating Schemas with the Repository Creation Utility*.

For more information about how to organize your schemas in a multi-domain environment, see "Planning Your Schema Creation" in *Creating Schemas with the Repository Creation Utility*.



Tip:

You must make a note of the custom prefix you choose to enter here; you will need this later on during the domain creation process.

Click **Next** to proceed, then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

3.1.3.5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords.

Tip:

You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

3.1.3.6 Specifying Custom Variables

Specify the custom variables for the SOA Infrastructure schema. For the Oracle Service Bus standard installation topology, accept both default values for the Database Profile (SMALL) and Healthcare Integration (NO).

Enter value for the following custom variables.

Component	Custom Variable	Value
SOA Infrastructure	Database Profile (SMALL/MED/LARGE)	SMALL
	Healthcare Integration(YES/NO)	NO

Tip:

For more information about the options on this screen, see "Custom Variables" in *Creating Schemas with the Repository Creation Utility*.

3.1.3.7 Completing Schema Creation

Navigate through the remainder of the RCU screens to complete schema creation. When you reach the Completion Summary screen, click **Close** to dismiss RCU.

For more information about RCU and its features and concepts, see *Creating Schemas with the Repository Creation Utility*

3.2 Configuring Your WebLogic Domain

This section provides instructions for creating a WebLogic domain using the configuration wizard.

For more information on other methods available for domain creation, see "Additional Tools for Creating, Extending, and Managing WebLogic Domains" in *Creating WebLogic Domains Using the Configuration Wizard*.

The following topics are covered in this section:

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Configure the Domain](#)

3.2.1 Starting the Configuration Wizard

To begin domain configuration, navigate to the `ORACLE_HOME/oracle_common/common/bin` directory and start the WebLogic Server Configuration Wizard.

On UNIX operating systems:

```
./config.sh
```

On Microsoft Windows operating systems:

```
config.cmd
```

On Windows operating systems, you can also start the Configuration Wizard from the **Start** menu by selecting **All Programs**, then selecting **Oracle**, then selecting **OracleHome**, then selecting **WebLogic Server 12c (12.1.3)**, then selecting **Tools**, then selecting **Configuration Wizard**.

3.2.2 Navigating the Configuration Wizard Screens to Configure the Domain

Follow the instructions in this section to create and configure the domain.

Note:

Extending an existing Service Bus domain with an API Manager domain is not supported in this release. You must create a new domain using the API Manager domain configuration template (which also deploys Service Bus) as described in this section.

- [Selecting the Domain Type and Domain Home Location](#)
- [Selecting the Configuration Templates](#)
- [Selecting the Application Home Location](#)
- [Configuring the Administrator Account](#)
- [Specifying the Domain Mode and JDK](#)
- [Specifying the Database Configuration Type](#)
- [Specifying JDBC Component Schema Information](#)

- [Testing the JDBC Connections](#)
- [Selecting Advanced Configuration](#)
- [Configuring the Administration Server Listen Address](#)
- [Configuring the Node Manager](#)
- [Configuring Managed Servers](#)
- [Configuring a Cluster](#)
- [Assigning Managed Servers to the Cluster](#)
- [Configuring Coherence Clusters](#)
- [Creating a New Machine](#)
- [Assigning Servers to Machines](#)
- [Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Writing Down Your Domain Home and Administration Server URL](#)

3.2.2.1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Create a New Domain**.

In the Domain Location field, specify your Domain home directory.

Oracle recommends that you locate your Domain home in accordance with the directory structure summarized in "What are the Key Oracle Fusion Middleware Directories?" in *Understanding Oracle Fusion Middleware*, where the Domain home is located outside the Oracle home directory. This directory structure will help you avoid issues when you need to upgrade or reinstall your software.

Tip:

For more information about the Domain home directory, see "Choosing a Domain Home" in *Planning an Installation of Oracle Fusion Middleware*.

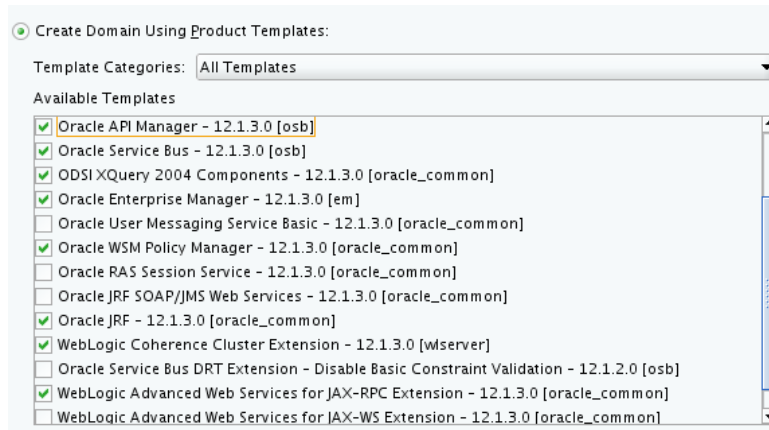
For more information about the other options on this screen, see "Configuration Type" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.2 Selecting the Configuration Templates

On the Templates screen, make sure **Create Domain Using Product Templates** is selected, then select the following templates:

- Oracle API Manager - 12.1.3.0 [osb]
Selecting this template also selects the following dependencies:
 - Basic WebLogic Server Domain - 12.1.3.0 [wlserver]
 - Oracle Service Bus - 12.1.3.0 [osb]
 - ODSI XQuery 2004 Components - 12.1.3.0 - [oracle_common]
 - Oracle Enterprise Manager - 12.1.3.0 [em]

- Oracle WSM Policy Manager - [oracle_common]
- Oracle JRF - 12.1.3.0 [oracle_common]
- WebLogic Coherence Cluster Extension - 12.1.3.0 [wlserver]
- WebLogic Advanced Web Services for JAX-PRC Extension - 12.1.3.0 [wlserver]

**Tip:**

For more information about the options on this screen, see "Templates" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.3 Selecting the Application Home Location

On the Application Location screen, select the location where you want to store your applications associated with your domain. This location is also referred to as the Application home directory.

Oracle recommends that you locate your Application home in accordance with the directory structure summarized in "What are the Key Oracle Fusion Middleware Directories?" in *Understanding Oracle Fusion Middleware*, where the Application home is located outside the Oracle home directory. This directory structure will help you avoid issues when you need to upgrade or reinstall your software.

Tip:

For more information about the Application home directory, see "Choosing an Application Home" in *Planning an Installation of Oracle Fusion Middleware*.

For more information about the options on this screen, see "Application Location" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.4 Configuring the Administrator Account

On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.

Oracle recommends that you make a note of the user name and password specified on this screen; you will need these credentials later to boot and connect to the domain's Administration Server.

3.2.2.5 Specifying the Domain Mode and JDK

On the Domain Mode and JDK screen:

- Select **Production** in the Domain Mode field.
- Select the **Oracle Hotspot JDK** in the JDK field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

Tip:

More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.

In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. For more information, see "Creating a Boot Identity File for an Administration Server" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

3.2.2.6 Specifying the Database Configuration Type

Select **RCU Data** to activate the fields on this screen. The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.

Note:

If you choose to select **Manual Configuration** on this screen, you will have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

After selecting **RCU Data**, fill in the following fields:

Field	Description
DBMS/Service	Enter the database DBMS name, or service name if you selected a service type driver.
Host Name	Enter the name of the server hosting the database.
Port	Enter the port number on which the database listens.
Schema Owner Schema Password	Enter the user name and password for connecting to the database's Service Table schema. This is the schema user name and password that was specified for the Service Table component on the "Schema Passwords" screen in RCU (see Creating the Database Schemas). The default user name is <i>prefix_STB</i> , where <i>prefix</i> is the custom prefix that you defined in RCU.

Specify AutoConfiguration Options Using:

RCU Data Manual Configuration

Enter the database connection details using the Repository Creation Utility service table (STB) schema credentials. The Wizard uses this connection to automatically configure the datasources required for components in this domain.

Vendor: Driver:

DBMS/Service: Host Name: Port:

Schema Owner: Schema Password:

Connection Result Log

Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Tip:

More information about the **RCU Data** option can be found in "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.

More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*

3.2.2.7 Specifying JDBC Component Schema Information

The JDBC component schemas associated with the products for which you are creating the domain are listed in the lower half of the screen.

Select the schemas for which you want to specify data source settings by selecting the check box adjacent to each schema name.

Fill in each field at the top of the screen with information for the database schema you created in [Creating the Database Schemas](#).

Field	Description
Vendor	Select the database vendor.
Driver	Select the JDBC driver to use to connect to the database. The list includes common JDBC drivers for the selected database vendor.
DBMS/Service	Enter a database DBMS name, or service name if you selected a service type driver.

Field	Description
Host Name	Enter the name of the server hosting the database.
Port	Enter the port number to be used to connect to the server that hosts the database.
Schema Owner	Enter the user name for connecting to the database's Oracle API Manager schema. This is the schema user name that was specified on the "Schema Passwords" screen in RCU (see Creating the Database Schemas). The default user name is <i>prefix_STB</i> , where <i>prefix</i> is the custom prefix that you defined in RCU.
Schema Password	Enter the password for connecting to the database's schema. This is the password that was specified for the Service Table component on the "Schema Passwords" screen in RCU (see Creating the Database Schemas).

Tip:

More information about the other options on this screen can be found in JDBC Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.8 Testing the JDBC Connections

Verify that the values on the JDBC Component Schema screen are correct for all schemas. If you selected **RCU Data** on the previous screen, the schema table should already be populated appropriately.

Tip:

More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.9 Selecting Advanced Configuration

To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- Administration Server
This is required to properly configure the listen address of the Administration Server.
- Node Manager
This is required to configure Node Manager.
- Managed Server, Clusters and Coherence
This is required to configure the Oracle API Manager Managed Server.

3.2.2.10 Configuring the Administration Server Listen Address

On the Administration Server screen, select the drop-down list next to Listen Address and select the IP address on the host where the Administration Server will reside. Do not use "All Local Addresses."

Do not specify any server groups for the Administration Server.

3.2.2.11 Configuring the Node Manager

The Node Manager screen can be used to select the type of Node Manager you want to configure, along with the Node Manager credentials.

Select **Per Domain Default Location** as the Node Manager type, then specify the Node Manager credentials.

Tip:

For more information about the options on this screen, see "Node Manager" in *Creating WebLogic Domains Using the Configuration Wizard*.

For more information about the types of Node Manager, see "Node Manager Overview" in *Administering Node Manager for Oracle WebLogic Server*.

3.2.2.12 Configuring Managed Servers

On the Managed Servers screen, a new Managed Server named `osb_server1` is created:

1. In the Listen Address drop-down list, select the IP address of the host on which the Managed Server will reside. Do not use All Local Addresses. Verify that **OSB-MGD-SVRS-COMBINED** is selected in the Server Group.

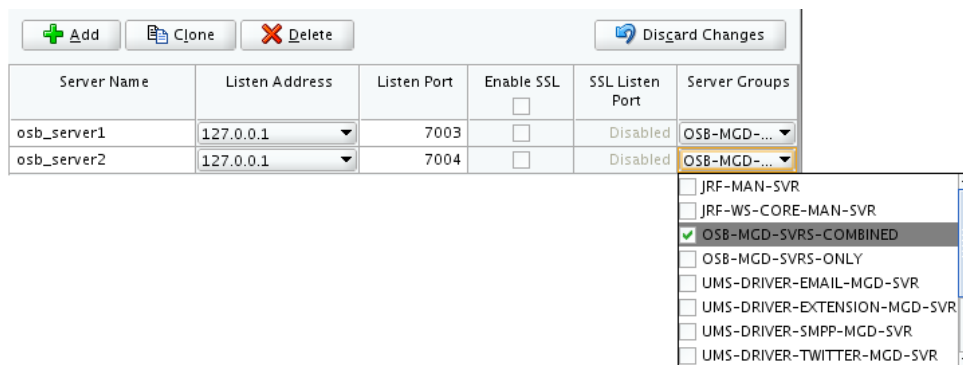
If you want to have OWSMPM in a different server from the Service Bus server, select **OSB-MGD-SVRS-ONLY**. This targets only Service Bus but not OWSMPM to the server.

2. Click **Add** and repeat this process to create a second Managed Server named `osb_server2`.

Configuring a second Managed Server is one of the steps needed to configure the standard topology for high availability. If you are not creating a highly available environment, then this step is optional.

For more information about the high availability standard topology, see "Understanding the Fusion Middleware Standard HA Topology" in *High Availability Guide*.

These server names will be referenced throughout this document; if you choose different names be sure to replace them as needed.



Tip:

For more information about the options on this screen, see "Managed Servers" in *Creating WebLogic Domains Using the Configuration Wizard*.

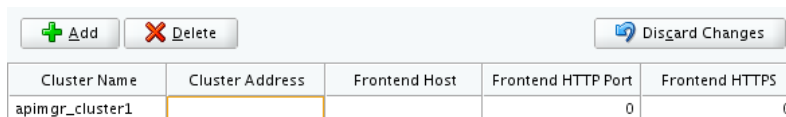
3.2.2.13 Configuring a Cluster

Use the Clusters screen to create a new cluster:

1. Click **Add**.
2. Specify `apimgr_cluster1` in the Cluster Name field.
3. Leave the Cluster Address field blank.

By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, see "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

New clusters can also be created using Fusion Middleware Control. In such cases, cluster communication (unicast or multicast) can be configured when the new cluster is created. For more information, see "Create and configure clusters" in *Oracle WebLogic Server Administration Console Online Help*.



Tip:

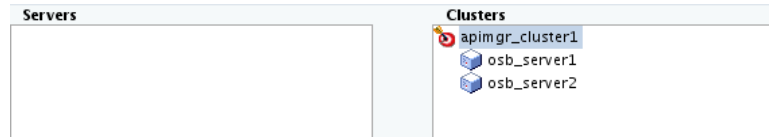
For more information about the options on this screen, see "Clusters" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.14 Assigning Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign `osb_server1` and `osb_server2` to the new cluster `apimgr_cluster1`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `apimgr_cluster1`.
2. In the Servers pane, assign `osb_server1` to `apimgr_cluster1` by doing one of the following:

- Click once on `osb_server1` to select it, then click on the right arrow to move it beneath the selected cluster (`apimgr_cluster1`) in the Clusters pane.
 - Double-click on `osb_server1` to move it beneath the selected cluster (`apimgr_cluster1`) in the clusters pane.
3. Repeat to assign `osb_server2` to `apimgr_cluster1`.

**Tip:**

For more information about the options on this screen, see "Assign Servers to Clusters" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.15 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the default port number 0 as the Coherence cluster listen port.

Note:

Setting the unicast listen port to 0 creates an offset for the Managed Server port numbers. The offset is 5000, meaning the maximum allowed value that can be assigned to a Managed Server port number is 60535, instead of 65535.

Note:

For Coherence licensing information, see Client Access to Oracle Coherence Clusters in *Oracle® Fusion Middleware Licensing Information User Manual*.

3.2.2.16 Creating a New Machine

Use the Machines screen to create a new machine in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.

Tip:

If you plan to create a high availability environment and know the list of machines required for your target topology, you can follow the directions in this section to create all of the machines at this time. For more information, see "Optional Scale Out Procedure" in *High Availability Guide*.

1. Click **Add** to create a new machine.
2. Specify `apimgr_machine1` in the Name field.
3. In the Node Manager Listen Address field, select the IP address of the machine where the Managed Servers are being configured.

You must select a specific interface and not "localhost." This allows Coherence cluster addresses to be dynamically calculated.

4. Verify the port in the Node Manager Listen Port field.

The port number 5556, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number as needed.

Note:

If you are extending an existing domain, you can assign servers to any existing machine. It is not necessary to create a new machine unless your situation requires it.

Name	Node Manager Listen Address	Node Manager Listen Port
apimgr_machine1	127.0.0.1	5556

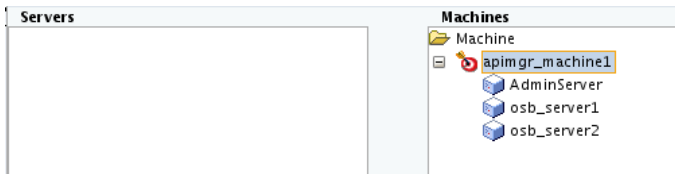
Tip:

For more information about the options on this screen, see "Machines" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.17 Assigning Servers to Machines

Use the Assign Servers to Machines screen to assign the Administration Server and Managed Servers to the new machine you just created:

1. In the Machines pane, select the machine to which you want to assign the servers; in this case, `apimgr_machine1`.
2. In the Servers pane, assign `AdminServer` to `apimgr_machine1` by doing one of the following:
 - Click once on `AdminServer` to select it, then click on the right arrow to move it beneath the selected machine (`apimgr_machine1`) in the Machines pane.
 - Double-click on `AdminServer` to move it beneath the selected machine (`apimgr_machine1`) in the Machines pane.
3. Repeat to assign both `osb_server1` and `osb_server2` to `apimgr_machine1`.



Tip:

For more information about the options on this screen, see "Assign Servers to Machines" in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.18 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.

You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

Domain creation will not begin until you click **Create**.

Tip:

More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

3.2.2.19 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL



Oracle Weblogic Server Configuration Succeeded

New Domain apidomain2_domain Creation Successful

Domain Location

[/MW_HOME/user_projects/domains/apidomain2_domain](#)

Admin Server URL

<http://11.222.333.444:7001/console>

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Administration Server, and the URL is needed to access the Administration Server.

Click **Finish** to dismiss the configuration wizard.

3.3 Starting the Servers

This section provides information about tools you can access to manage your domain, after the configuration is complete.

Note:

For more information on additional tools you can use to manage your domain, see "Overview of Oracle Fusion Middleware Administration Tools" in *Administering Oracle Fusion Middleware*.

- [Starting the Node Manager](#)
- [Starting the Administration Server](#)
- [Starting the Managed Servers](#)

3.3.1 Starting the Node Manager

To start your per-domain Node Manager, go to the `DOMAIN_HOME/bin` directory.

On UNIX operating systems, start the Node Manager as shown below, using `nohup` and `nm.out` as an example output file:

```
nohup ./startNodeManager.sh > $LOG_DIR/nm.out&
```

In the preceding command, `LOG_DIR` is the location of directory in which you want to store the log files.

On Windows operating systems, run:

```
startNodeManager.cmd
```

Note:

On Windows operating systems, it is recommended that you configure Node Manager to run as a startup service. This allows Node Manager to start up automatically each time the system is restarted.

For more information, see "Running Node Manager as a Startup Service" in *Administering Node Manager for Oracle WebLogic Server*.

For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

3.3.2 Starting the Administration Server

To start the Administration Server, go the `DOMAIN_HOME/bin` directory.

On UNIX operating systems, run:

```
./startWebLogic.sh
```

On Windows operating systems, run:

```
startWebLogic.cmd
```

If you have selected **Production Mode** on the Domain Mode and JDK screen in [Specifying the Domain Mode and JDK](#), you will be prompted for the login credentials of the Administrator user as provided on the Administrator Account screen in [Configuring the Administrator Account](#).

Tip:

For more information about starting the Administration Server, see "Starting and Stopping Administration Servers" in *Administering Oracle Fusion Middleware*.

In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. For more information, see "Creating a Boot Identity File for an Administration Server" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

You can verify that the Administration Server is up and running by accessing the Administration Server Console. The URL is provided on the Configuration Success screen in [Writing Down Your Domain Home and Administration Server URL](#).

Note:

Make sure that the database hosting your product schemas is up and running and accessible by the Administration Server.

`http://administration_server_host:administration_server_port/console`

The default Administration Server port number is 7001.

For more information about how to use the Administration Console, see "Getting Started Using Oracle WebLogic Server Administration Console" in *Administering Oracle Fusion Middleware*.

3.3.3 Starting the Managed Servers

To start the Managed Servers:

1. Log in to Oracle Fusion Middleware Control:

`http://administration_server_host:administration_server_port/em`

The Administration Server host and port number were in the URL on the Configuration Success screen ([Writing Down Your Domain Home and Administration Server URL](#)). The default Administration Server port number is 7001.

The login credentials were provided on the Administrator Account screen ([Configuring the Administrator Account](#)).

2. From the "Target Navigation" pane, click the arrows to expand the domain until the Managed Servers (osb_server1 and osb_server2) are visible.
3. Select the first Managed Server (osb_server1).
4. Next to the **WebLogic Server** menu, select **Start Up**.
5. Repeat Steps 3 and 4 to start osb_server2.
6. In the Target Navigation page, select your domain name to see that all the servers are up and running.

3.4 Next Steps

The next steps include creating users and assigning user roles.

See [Managing Users in Oracle API Manager](#) for more information.

Managing Users in Oracle API Manager

This chapter describes managing users in Oracle API Manager.

The following topics are covered:

- [Managing Users](#)
- [Next Steps](#)

4.1 Managing Users

API Manager users are managed using Fusion Middleware Control. An administrative user creates groups in Fusion Middleware Control, maps application roles to these groups, creates users, and then assigns users to groups.

Use this URL to log onto Fusion Middleware Control:

```
http://administration_server_host:administration_server_port/em
```

The Administration Server host and port number were in the URL on the Configuration Success screen ([Writing Down Your Domain Home and Administration Server URL](#)). The default Administration Server port number is 7001.

Note:

The tasks in this section describe creating users and assigning roles using Fusion Middleware Control. You can also use an LDAP Authentication provider to manage users. See "Configuring LDAP Authentication Providers" in *Administering Security for Oracle WebLogic Server* for more information.

4.1.1 Create Groups for API Manager Roles

Use Fusion Middleware Control to create groups that map to the API Manager roles.

See [Understanding User Roles in Oracle API Manager](#) for additional information about the API Manager user roles.

You must complete this task multiple times to create the following groups:

- API Admin
- API Consumer
- API Curator
- Developer

You do not need to create groups for the following Service Bus groups, which are available OOTB:

- Administrators
- Deployers
- Monitors

To create a group:

1. Log in to Fusion Middleware Control as a user with administrator privileges.
2. In the Target Navigator, expand **WebLogic Domain**, and right-click the name of your domain. Navigate to **Security > Users and Groups** to display the Users and Groups page.
3. Click the **Groups** tab.
4. Click **Create**.
5. From the Create a New Group page, define the following properties for the groups listed above:
 - Name (must be unique)
 - Description
 - Provider
6. Click **Create**.
7. Repeat steps 4 through 6 to create the remaining groups.

When finished, complete the task described in [Assign Application Roles to Groups Using Fusion Middleware Control](#).

4.1.2 Assign Application Roles to Groups Using Fusion Middleware Control

After you have created groups that correspond with the roles in API Manager, you must assign application roles to these groups. After completing this task, any users assigned to the specified groups will be granted the applicable application role.

To assign application roles to groups:

1. Log in to Fusion Middleware Control as a user with administrative privileges.
2. In the Target Navigator, expand **WebLogic Domain**, and right-click the name of your domain. Navigate to **Security > Application Roles** to display the Application Roles page.
3. Select **Service_Bus_Console** from the Application Stripe list, and then click the **Search Application Roles** icon.
4. Map the API Curator application role to the API Curator group:
 - a. Select **APICurator** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.

- c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **API Curator** group, and then click **OK** to close the dialog.
 - e. Click **OK**.
5. Map the Developer application role to the API Developer group:
 - a. Select **Developer** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.
 - c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **API Developer** group, and then click **OK** to close the dialog.
 - e. Click **OK**.
6. Map the Deployer application role to the Deployers group:
 - a. Select **Deployer** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.
 - c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **Deployers** group, and then click **OK** to close the dialog.
 - e. Click **OK**.
7. Map the Monitor application role to the Monitors group:
 - a. Select **Monitor** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.
 - c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **Monitors** group, and then click **OK** to close the dialog.
 - e. Click **OK**.
8. Select **API_Manager** from the Application Stripe list, and then click the **Search Application Roles** icon.
9. Map the API Admin application role to the API Admin group:
 - a. Select **APIApplicationAdminsitrator** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.
 - c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **API Admin** group, and then click **OK** to close the dialog.

- e. Click **OK**.
10. Map the API Consumer application role to the API Consumer group:
- a. Select **APIConsumer** from the list of application roles, and then click **Edit**.
 - b. From the Members region, click **Add**.
 - c. From the Add Principal dialog, select **Group** from the Type list, and then click the **Search** icon.
 - d. Select the **API Consumer** group, and then click **OK** to close the dialog.
 - e. Click **OK**.

You do not need to assign application roles to the Administrator Service Bus group. This is done OOTB.

When finished, complete the task described in [Creating API Manager Users](#).

4.1.3 Creating API Manager Users

You create API Manager users with Fusion Middleware Control.

You should create at least one user for each of the following roles:

- API Admin
- API Consumer
- API Curator
- API Developer
- Deployers
- Monitors

Caution:

Do not use any of the following characters in user names: ; , + = \ (double back-slashes can be used; for example `smith\\`). Do not begin a user name with a pound sign (#) or double quotes ("). Creating a user with any of the preceding invalid characters can corrupt the WebLogic domain.

To create API Manager users:

1. Log in to Fusion Middleware Control as a user with administrator privileges.
2. In the Target Navigator, expand **WebLogic Domain**, and right-click the name of your domain. **Navigate to Security > Users and Groups** to display the Users and Groups page
3. Click the **Users** tab.
4. Above the Users table click **Create**.
5. In the **Name** field of the Create New User dialog enter the login ID of the user.

6. Optionally, in the **Description** field, enter a short description to help identify the user.
7. In the **Provider** drop-down list, select the authentication provider for the user.
8. In the **Password** field, enter a password for the user. The password must be 8 characters or more.
9. Re-enter the password for the user in the **Confirm Password** field.
10. Click **Create** to save your changes.
11. Repeat steps 4 through 10 to create the remaining users.

The user name appears in the User table

When finished, complete the task described in [Assigning Users to Groups](#).

4.1.4 Add Groups to the Monitors Parent Group

All users accessing the Service Bus console must be a member of the Monitors group. The easiest way to assign these users to the Monitors group is to add their parent groups to the Monitors group.

You must add these groups to the Monitors group:

- API Curator
- API Developer
- Deployers

Note:

All users accessing the Service Bus Console must be added to the Monitors parent group or to a group that is a member of the Monitors parent group. Ensure that you have completed this task if you are having trouble accessing the Service Bus Console with an appropriate user.

To add groups to the Monitors parent group:

1. Log in to Fusion Middleware Control as a user with administrator privileges.
2. In the Target Navigator, expand **WebLogic Domain**, and right-click the name of your domain. Navigate to **Security > Users and Groups** to display the Users and Groups page.
3. Click the **Groups** tab.
4. Click the group you want to add to Monitors. As an example, click **API Curator**.
5. Click the **Membership** tab.
6. Select the **Monitors** group from the Available list, and then click the **Right Arrow (>)** icon to move the **Monitors** group to the Chosen list.
7. Click **Save**, and then click the **Users and Groups** link to return to the Users and Groups page.

- Repeat steps 4 through 7 for each of the remaining groups.

When finished, complete the task described in [Assigning Users to Groups](#).

4.1.5 Assigning Users to Groups

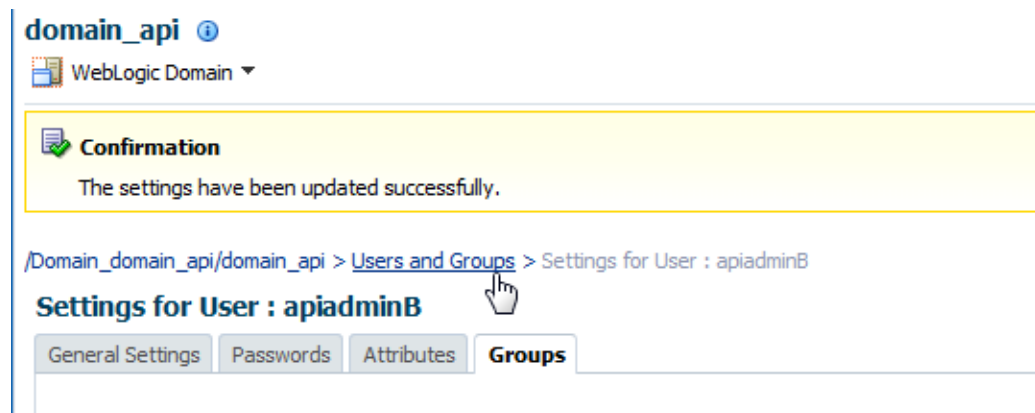
You add users to the appropriate groups to grant role permissions associated with that group. For example, assign a user to the API Curator group to grant that user permissions associated with the API Curator role.

You should create at least one user for each role, and then add users to the groups that correspond with their intended roles:

- API Admin
- API Consumer
- API Curator
- API Developer
- Deployers
- Monitors

To add API Manager users to groups:

- Log in to Fusion Middleware Control as a user with administrator privileges.
- In the Target Navigator, expand **WebLogic Domain**, and right-click the name of your domain. Navigate to **Security > Users and Groups** to display the Users and Groups page.
- Click the **Users** tab.
- In the Users table, click the name of the user you created in [Creating API Manager Users](#).
- From the Setting for User page, click the **Groups** tab.
- Select the groups to which you want to add the user and then click the right arrow to add them to the Chosen list. As an example, an API Curator user should be added to the API Curator group.
- Click **Save**, and then click the Users and Groups links, as shown in the following figure, to return to the Users and Groups page.



8. Repeat steps 4 through 7 for each user you created in [Creating API Manager Users](#).

4.2 Next Steps

The next steps include curating APIs using Oracle Service Bus, discovering and using APIs from the API Manager Portal, and administering API Manager.

See [Using Oracle API Manager](#) for more information.

Part III

Using Oracle API Manager

This section describes using Oracle API Manager, including curating APIs with Oracle Service Bus, discovering APIs, using the API Manager Portal, and administering API Manager.

This section contains the following chapters:

- [Creating your APIs](#)
- [Curating and Publishing Your APIs Using Oracle Service Bus](#)
- [Using the Oracle API Manager Portal](#)
- [Administering Oracle API Manager](#)

Creating Your APIs

This chapter describes creating APIs using the Oracle Service Bus console.

The following topics are covered:

- [Accessing the Oracle Service Bus Console](#)
- [Creating Proxy Services to Be Published as APIs](#)
- [Activating the Session](#)
- [Next Steps](#)

5.1 Accessing the Oracle Service Bus Console

You access the Oracle Service Bus Console from a web browser.

To start the Oracle Service Bus Console:

1. Make sure the API Manager domain and all servers are running. If the servers are down, an administrative user can start them by completing the tasks described in [Starting the Servers](#).
2. When the servers are started, enter one of the following URLs in your browser:

```
http://hostname:port/servicebus
```

or, for SSL:

```
https://hostname:port/servicebus
```

where **hostname** represents the hostname of the API Manager Admin Server and **port** represents the port number on which it is listening.

The login page appears.

3. Enter the username and password of a user with the appropriate role, and then click **Sign In**.

5.2 Creating Proxy Services to Be Published as APIs

WSDL-based SOAP proxy services and REST proxy services created in Oracle Service Bus are able to be published as APIs to the API Manager Portal. You use Service Bus capabilities to create proxy services before adding decorative metadata and publishing the proxy service as an API on the API Manager Portal.

Note:

Proxy services configured with deprecated WLS 9 policies can not be published as APIs.

See *Developing Services with Oracle Service Bus* for more information about developing services with Service Bus.

You can also import a configuration JAR containing proxy services from an existing Service Bus instance. See [Importing and Exporting APIs and API Curation Metadata](#) for more information.

Note:

You still must add metadata and publish proxy services added using this method.

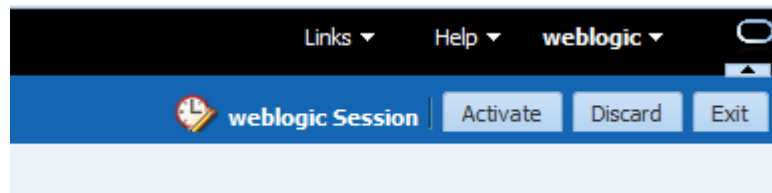
After populating Service Bus with proxy services, please ensure that you activate the session. After activating the session, select a SOAP- or REST-based proxy service in the Navigator to view the API tab. Note that the API tab is only visible for SOAP- and REST-based proxy services. Complete the task described in [Activating the Session](#) to make the API tab visible in the Service Bus console.

5.3 Activating the Session

After you have created your API proxy service, you must activate the session before adding metadata to and publishing an API for use with the API Manager Portal. The API tab is not visible in the Service Bus console until you complete this task.

To activate the session:

1. Click **Activate**, as shown in the following image.



2. From the Confirm Session Activation dialog:
 - a. **(Optional)** Type a description of the changes made in this session into the **Description** field.
 - b. Click **Activate** to activate the session.

The session is activated. The API tab is now available in the Service Bus console. .

Note:

Even though the service has been activated, it will not receive any traffic until it is published. You must curate and publish your APIs, as described in [Curating and Publishing Your APIs Using Oracle Service Bus](#), before the services will receive traffic.

5.4 Next Steps

Next steps include adding metadata and publishing APIs to the API Manager Portal.

See [Curating and Publishing Your APIs Using Oracle Service Bus](#) for more information.

Curating and Publishing Your APIs Using Oracle Service Bus

After creating your APIs, you must add metadata to and publish the API before it receives any traffic. This chapter describes curating and publishing WSDL-based SOAP and REST proxy services as APIs using the Oracle Service Bus Console, which is installed with API Manager.

The tasks described here are performed by users with the API Curator role.

Note:

API proxy services need to be created and activated before you begin the tasks in this chapter. See [Creating Your APIs](#) for more information.

The following topics are covered:

- [Editing the Metadata of an API](#)
- [Understanding API Status](#)
- [Making an API Private](#)
- [Publishing an API to the Oracle API Manager Portal](#)
- [Hiding an API from the API Manager Portal by Changing Its Status Back to Draft](#)
- [Deprecating an API](#)
- [Next Steps](#)

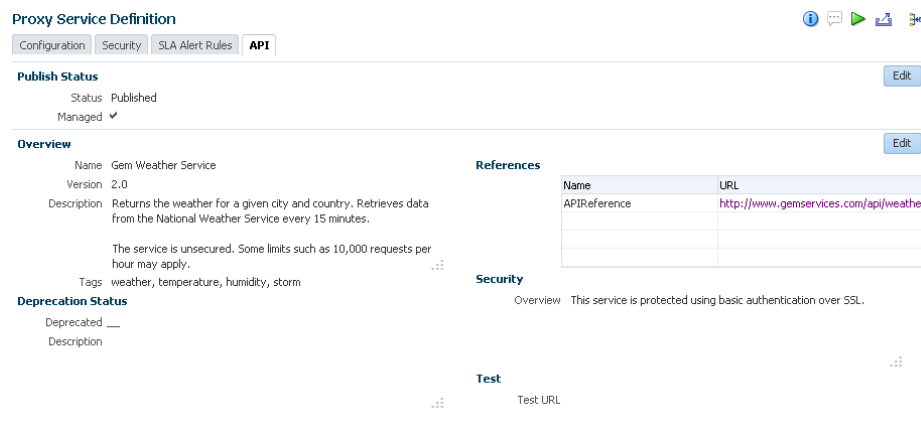
6.1 Editing the Metadata of an API

After creating a proxy service, you can add descriptive metadata that is displayed in an API detail page in the API Manager Portal.

API Consumers use this metadata to determine if an API is a good choice for their application and to test the API. Users with access to the Service Bus console can also evaluate this API using the metadata.

To edit the metadata of an API:

1. From the Oracle Service Bus Console, open the Proxy Service Definition for the service you want to publish.
2. Click the **API** tab. From the Overview region, click **Edit**, as shown in the following figure.



3. Edit the metadata properties, as described in the following table:

Element	Description
Overview	Use this section to modify basic metadata associated with an API.
Name	Enter a name for the API
Version	Enter version information for the proxy service.
Description	Enter a detailed description of the functionality of an API.
Tags	Enter any tags you want to associate with the API. You can search for APIs using these tags in the Oracle API Manager portal.
Deprecation Status	Use this section to set options related to deprecating an API.
Deprecated	Select to deprecate the API.
Description	Enter text describing why the API was deprecated, including superceding versions, if any.
References	Use this section to add external documentation references or other information sources. Note: You are limited to a maximum of four references.
Name	Enter a name for the reference.
URL	Enter the URL for the reference.
Security	Use this section to add security information.
Overview	Enter an overview of the global security policies attached to this API. Note: Descriptions of individual security policies from OWSM are exposed to consumers on the API Details page.
Test	Use this section to add testing information.
Test URL	Enter a URL used to test the API.

Tip:

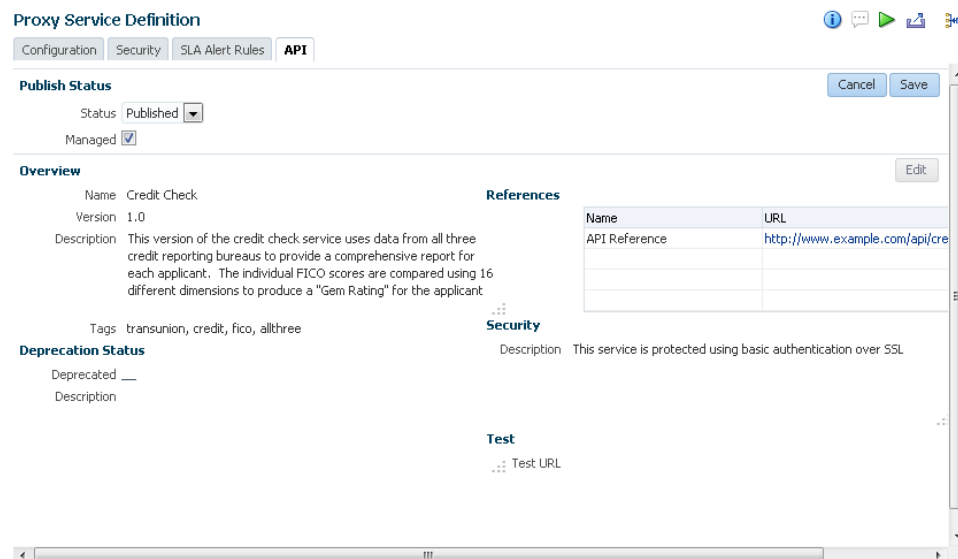
You can include usage instructions for the Test URL in the **Description** field in the **Overview** section or as an HTML reference in the **Reference** section.

4. Click **Save**. The metadata for the API is now updated to reflect your changes in both the Oracle Service Bus Console and the API Manager Portal.

6.2 Understanding API Status

The status of an API determines whether it is hidden from or published to the API Manager Portal.

API status is set in the Status list in Publish Status region on the API tab, as shown in the following figure:



You can set the following status for an API:

- **Draft:** Draft APIs are not published to the Oracle API Manager Portal and all requests to them are blocked. This is the default option.
- **Private:** Private APIs are available for users with Developer privileges in Oracle Service Bus. Private APIs are not visible in the API Manager Portal and are unavailable to users with the API Consumer role. [Making an API Private](#) describes making an API private.
- **Published:** Published APIs are published to consumers on the Oracle API Manager Portal. [Publishing an API to the Oracle API Manager Portal](#) describes publishing an API.

6.3 Making an API Private

You can make an API private if you want it to be available for users with Developer privileges in Service Bus. APIs marked private are not visible in the API Manager Portal and are unavailable to users with the API consumer role.

To make an API private:

1. From the Oracle Service Bus Console, open the Proxy Service Definition for the service you want to make private.

2. Click the **API** tab. From the Publish Status region, click **Edit**.
3. From the Status list, select **Private**.
4. Click **Save**. The API is now private.

6.4 Publishing an API to the Oracle API Manager Portal

Before making an API available to consumers through the API Manager Portal, you should decide whether it should be managed or unmanaged. You then change the status to Published to publish the API.

Tip:

Make sure that you have added metadata to the API, as described in [Editing the Metadata of an API](#), before publishing it to the API Manager Portal.

6.4.1 Deciding if an API Should Be Managed or Unmanaged

Before publishing an API, you decide whether an API should be published as managed or unmanaged:

- Managed APIs can only be invoked by passing the application's entitlement key as a header at runtime. Users must subscribe to an API to receive this key. Users attempting to invoke managed APIs without using the key will receive an error message.
- Unmanaged APIs are not controlled at runtime and can be accessed without passing an entitlement key. Users can invoke these APIs without subscribing to them.

You select the **Managed** check box when you publish an API to make it managed, as described in [Changing the Status of an API to Published](#). If you later decide that the API should not be published, deselect the **Managed** check box.

Note:

If you decide to make a previously managed API unmanaged, user requests to the API that include an application's entitlement key, which is required for managed but not unmanaged APIs, will ignore the `X-API-KEY` attribute in the header. The request is processed normally.

6.4.2 Changing the Status of an API to Published

You change the status of an API to published to publish it to the API Manager Portal.

To publish an API:

1. From the Oracle Service Bus Console, open the Proxy Service Definition for the service you want to publish.
2. Click the **API** tab. From the Publish Status region, click **Edit**.
3. From the Status list, select **Published**.
4. Select the **Managed** check box if you want this API to be managed.
5. Click **Save**. The API is published to the API Manager portal.

Note:

Clicking the **Clear All** button in this region resets all fields to their default values. You must then click **OK** to confirm.

6.5 Hiding an API from the API Manager Portal

If you have published an API, but later decide that it should not be visible on the API Manager Portal and all requests to the API should be blocked, then you can set its status back to draft.

Note:

Deleting a proxy service from the Service Bus Console will also remove it from the API Manager Portal.

To change the status of an API to draft:

1. From the Oracle Service Bus Console, open the Proxy Service Definition for the service you want to mark as Draft.
2. Click the **API** tab. From the Publish Status region, click **Edit**.
3. From the Status list, select **Draft**.
4. Click **Save**. The API is now hidden from the API Manager Portal.

6.6 Deprecating an API

Deprecated APIs will remain published to the API Manager Portal. Users can view the details of, but cannot subscribe to, deprecated APIs.

You can provide details about why the API was deprecated, such as which API supersedes the deprecated version.

Note:

Users who have subscribed to an API before it is deprecated will still be subscribed to it. Requests to the API are processed normally.

To deprecate an API:

1. From the Oracle Service Bus Console, open the Proxy Service Definition for the service you want to deprecate.
2. Click the **API** tab. From the Overview region, click **Edit**.
3. From the Deprecation Status section, select the **Deprecated** option.
4. From the Deprecation Status section, enter a deprecation description into the **Description** field, as shown in the following figure. This description should include the reason the API is deprecated and any superceding versions.

Deprecation Status

Deprecated

Description

API is deprecated in Favor of the Validate Credit 2.0 API

5. Click **Save**. The API is now marked as Deprecated in the API Manager Portal.

6.7 Next Steps

Next steps include discovering, testing, and using APIs.

See [Using the Oracle API Manager Portal](#) for more information.

Using the Oracle API Manager Portal

This chapter describes discovering and subscribing to APIs using Oracle API Manager. After APIs have been published to the Oracle API Manager Portal, users with the API Consumer role can use the portal to find and subscribe to APIs for use in their applications.

The following topics are covered:

- [Introducing the Oracle API Manager Portal](#)
- [Accessing the API Manager Portal](#)
- [Managing Applications](#)
- [Discovering APIs](#)
- [Viewing Your Subscriptions](#)
- [Using APIs in Your Applications](#)
- [Unsubscribing from APIs](#)
- [Next Steps](#)

7.1 Introducing the Oracle API Manager Portal

Oracle Service Bus proxy services exposed as APIs are published to the Oracle API Manager Portal. From the API Manager Portal, users with the API Consumer role can discover, research, subscribe to, and use APIs.

Users manage their subscriptions using the Subscriptions page and discover and research new APIs using the Catalog page.

Users also use the API Manager Portal to find the necessary information to use the APIs with their applications. APIs are consumed in the context of applications. Each application for which users intend to use published APIs must be represented in the API Manager Portal. This allows for the management of APIs and for the collection of subscription statistics for each application.

Users with the API Admin role can perform additional tasks on the API Manager Portal. See [Administering Oracle API Manager](#) for information about these tasks.

7.2 Accessing the API Manager Portal

This section describes signing in to and logging out of the API Manager Portal.

7.2.1 Signing in to the API Manager Portal

To sign in to the API Manager Portal:

1. Navigate to the Oracle API Manager portal URL, for example: `http://example.com:7001/apimanager/`
2. Enter your user name and password into the fields, and then click **Sign In**.

7.2.2 Logging Out of the API Manager Portal

To Log out of the API Manager Portal, click your user name to expand the user menu, and then click **Logout**, as shown in the following figure.



7.3 Managing Applications

Each application with which users plan to use APIs must be represented in the API Manager Portal.

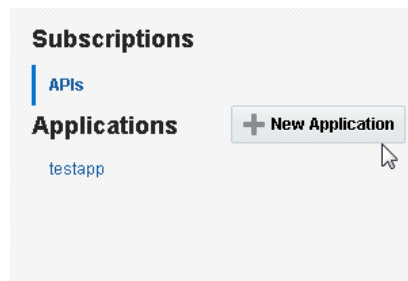
This section describes creating applications, subscribing APIs to an application, and removing applications that are no longer used from the API Manager Portal.

7.3.1 Creating an Application

Before you can subscribe APIs to an application, an application must be created in the API Manager Portal that corresponds to the application with which you plan to use the API.

To create an application from the Subscriptions page:

1. From the Applications panel, click the **New Application** button, as shown in the following figure.



2. In the **Name** field, enter a unique name for the application.
3. In the **Description** field, enter a description for the application.
4. Click **Create** to create the application.

Note:

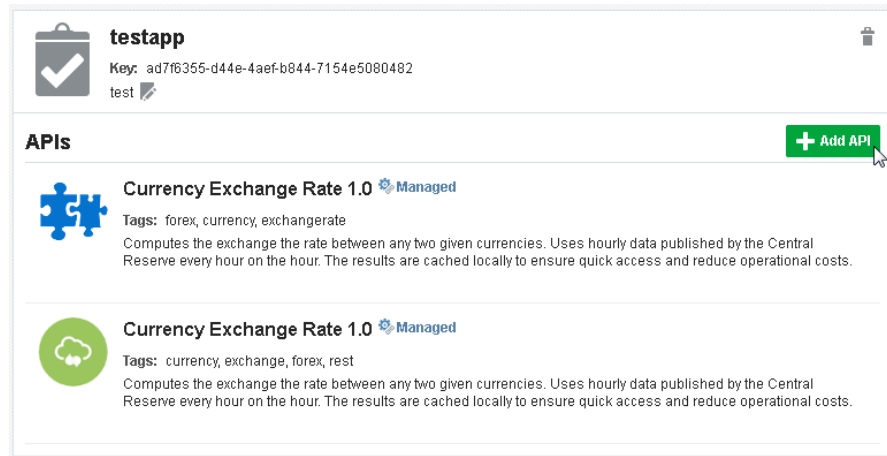
You can also create an application from the Catalog page. See [Subscribing to APIs from the Catalog Page](#), for more information.

7.3.2 Subscribing to APIs from the Subscriptions Page

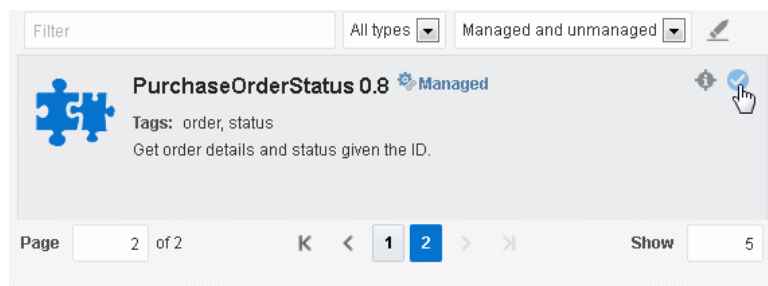
After you have created an application, you can subscribe to APIs for use with the application from the Subscriptions page.

To subscribe APIs to an application from the Subscriptions Page:

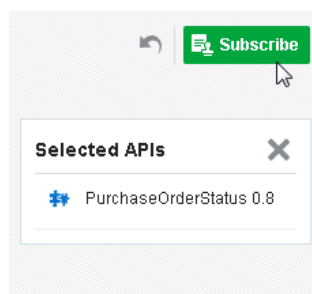
1. From the Subscriptions page, click the name of an application to display it in the Application panel.
2. From the APIs region, click the **Add APIs** button, as shown in the following figure.



3. Find APIs to which you want to subscribe to. See [Finding APIs Using the Catalog Page](#), for more information about finding APIs.
4. For each API you want to subscribe to, click the **Select** icon, as shown in the following figure.



5. From the Selected APIs region, click the **Subscribe** button, as shown in the following figure.



Note:

You can remove selected APIs by hovering over an API and clicking the **Unselect** icon next to an API in the Selected APIs list.

You can remove all selected APIs by clicking the **Clear** icon in the Selected APIs region.

7.3.3 Editing an Application Description

You can edit an application description to change its description in the API Manager Portal.

To edit an application description:

1. From the Subscriptions page, click the name of the application for which you want to edit the description.
2. Click the **Edit** icon, as shown in the following figure.



3. Edit the description in the text area, and then click the **Save** icon to save or the **Cancel** icon to cancel.

7.3.4 Removing Applications from the API Manager Portal

You can remove an application from the API Manager Portal if the application is no longer being used. The APIs will remain published in the API Manager Portal, but calls to managed APIs using the application's old entitlement key will be rejected.

To remove an application from the API Manager Portal:

1. Navigate to the Subscriptions Page.
2. Click the name of the application you want to delete to display its Application panel.
3. Click the **Delete** icon, as shown in the following figure.



4. When prompted, click **Yes** to confirm deletion.

7.4 Discovering APIs

Users with the API Consumer role can use the API Manager Portal to search for APIs and to review the metadata associated with APIs of interest to determine whether the API would be of use in their applications.

Users can subscribe to APIs that are not deprecated directly from the Catalog page.

7.4.1 Finding APIs Using the Catalog Page

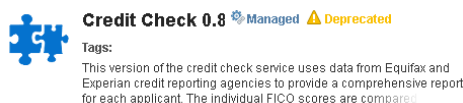
You use the Catalog page of the API Manager Portal to search for APIs to use in your applications. From the Catalog page, you can view the details of each API to decide if you want to use it in your application.

To find APIs:

1. From the Catalog page, enter search text into the **Filter** field. The search returns APIs with a Name, Description, or Tags that matches this text.

Note:

Deprecated APIs cannot be selected. They appear with a Deprecated symbol, as shown in the following figure.



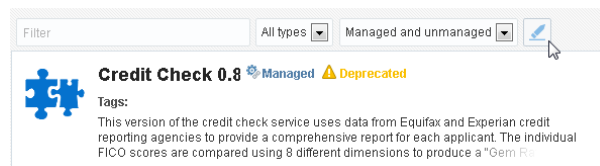
You can hover over the Deprecated symbol to read the reason the API is deprecated.

2. From the Type list, select the type of APIs to be returned. Select **All types** to return APIs of all types (this option is selected by default), select **SOAP** to return WSDL-based SOAP APIs, or select **REST** to return REST APIs.
3. From the Managed list, select the managed status of APIs to be returned. Select **Managed and unmanaged** to return both managed and unmanaged APIs, select **Managed** to return only managed APIs, or select **Unmanaged** to return only unmanaged APIs.

[Understanding the Difference Between Managed and Unmanaged APIs](#), describes the difference between managed and unmanaged APIs.

Tip:

Click the **Clear Filter** icon, as shown in the following figure, to clear the search results and return the complete list of APIs.



7.4.2 Viewing Details of an API

After you have discovered an API you are interested in, you can view its Details page to review its associated metadata.

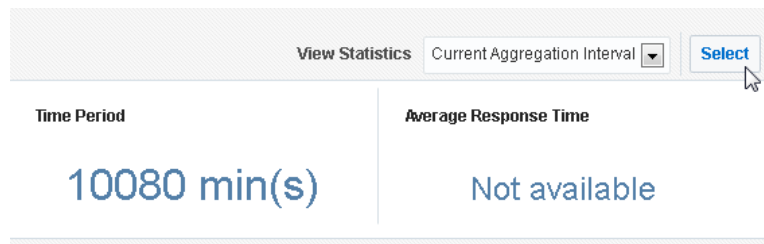
To view an API details page:

1. Find an API using the task described in [Finding APIs Using the Catalog Page](#).

- Click the **Details** icon, as shown in the following figure, for an API to open its Details page. Data displayed on this page is described in [Data Displayed on API Detail Pages](#).



- (Optional) To subscribe to this API, click **Select** to add this API to the Selected APIs region of the Catalog page, as shown in the following figure and then complete the task described in [Subscribing to APIs from the Catalog Page](#) (starting with Step 3) to complete the subscription process.



7.4.3 Data Displayed on API Detail Pages

The API Detail page displays the following status for an API:

- Managed Status:** Indicates whether the API is managed or unmanaged.
- Deprecation Status:** Indicates whether the API is deprecated or undeprecated

The API Detail page displays the following analytics:

- Current API Status:** Indicates whether the API is up or down.
- Error Count:** Displays the number of errors received when invoking this API.
- Message Count:** Displays the number of messages received from the API.
- Time Period:** Displays the length of time for which the displayed data has been aggregated.
- Average Response Time:** Displays the average message response time (in milliseconds). See "Configuring Operational and Global Settings" in *Administering Oracle Service Bus* for more information about changing the aggregation interval.

The analytics displayed are determined by the option selected in the View Statistics list. Select **Current Aggregation Interval** to display analytics for only the current aggregation interval (the Time Period displayed on the detail page), or select **Since Last Reset** to display analytics statistics since the last reset.

The API Detail page also displays the following metadata:

Element	Description
API Profile	Displays basic metadata associated with an API.

Element	Description
API URL	Links to the URL of the API.
Version	Displays version information for the API
Description	Contains a detailed description of the functionality of an API.
Tags	Lists tags associated with the API.
Management	Indicates whether the API is managed or unmanaged.
SOAP	Displays links for the associated WSDL file. Note: this region is displayed only for WSDL-based SOAP APIs.
WSDL	Displays a link to the WSDL file.
OraWSDL	Displays a link to the OraWSDL file, if applicable.
REST	Displays a link to the WADL file Note: This region is displayed only for REST APIs.
References	Displays external documentation references and other information sources for the API.
API Details	Displays additional details for the API.
Security Overview	Displays an overview of the Global security policies attached to this API.
Security Policies	Displays the security policies attached to this API. Hover over a policy and click its Details icon to display a policy's description.
Deprecation	Indicates whether the API is deprecated or not deprecated.
Testing	Displays the testing URL for the API.

7.4.4 Subscribing to APIs from the Catalog Page

You can subscribe to APIs directly from the Catalog page.

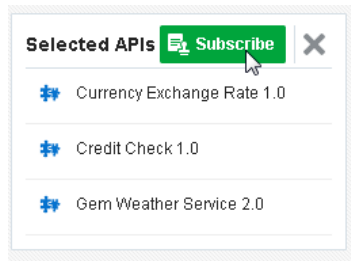
To subscribe to APIs from the Catalog page:

1. From the Catalog page, click the **Select** icon for each API to which you want to subscribe. You can select multiple APIs for subscription.

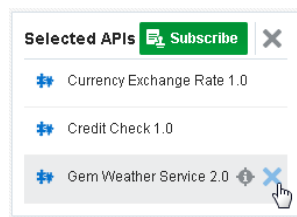
Note:

You cannot select deprecated APIs. They are not available for new subscriptions.

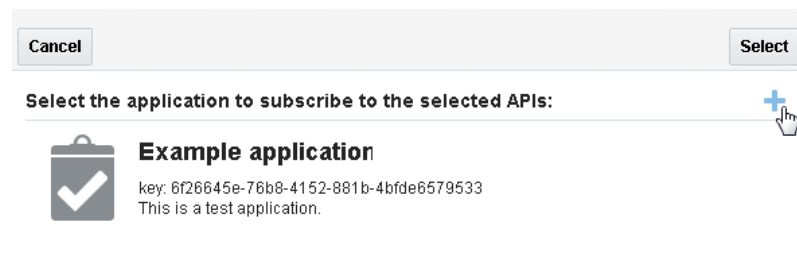
2. From the Selected APIs region, click the **Subscribe** button, as shown in the following figure.

**Note:**

You can remove selected APIs by clicking the **Unselect** icon next to an API in the Selected APIs list. You can remove all selected APIs by clicking the **Clear** icon in the Selected APIs region. These icons are displayed in the following figure.



3. (Optional) If the application you want to use is not listed:
 - a. Click the **Create** icon, as shown in the following figure.



- b. From the Create Application dialog, enter a name for the application into the **Name** field, and then enter a description into the **Description** field.
 - c. Click the **Create** button to create the application.
4. Click an application to select it, and then click the **Select** button. You are subscribed to the selected APIs.

7.5 Viewing Your Subscriptions

You can view the APIs you have subscribed to using two different views.

- Use the APIs panel of the Subscriptions page to display a list of APIs to which you are subscribed. This is described in [Viewing Your Subscribed APIs](#).
- Use the Applications panel of the Subscriptions page to view the APIs to which a selected application is subscribed. This is described in [Viewing APIs Subscribed to a Specific Application](#).

7.5.1 Viewing Your Subscribed APIs

To view your subscribed APIs, from the Subscriptions page, click **APIs**. Each API subscribed to appears in the list. The Name, Tags, Application, Key, and Description are displayed for each API.

Tip:

You can use the **Order by** list to sort APIs based on the following criteria:

- API name, descending
- API name, ascending
- application name, descending
- application name, ascending

If multiple applications are subscribed to the same API, that API will be listed multiple times, once for each application that is subscribed to it, as shown in the following figure.

The screenshot shows a list of APIs with the following details:

API Name	Managed	Tags	Application	Key	Description
Currency Exchange Rate	Managed	currency, exchange, forex, rest	testapp	ad7f6355-d44e-4aef-b844-7154e5080482	Computes the exchange the rate between any two given currencies. Uses hourly data published by the Central Reserve every hour on the hour.
Gem Weather Service	Managed	weather, temperature, humidity, storm	testapp	ad7f6355-d44e-4aef-b844-7154e5080482	Returns the weather for a given city and country. Retrieves data from the National Weather Service every 15 minutes. The service is unsecured.
Gem Weather Service	Managed	weather, temperature, humidity, storm	Example Application	8a65e01c-5675-43de-9d47-822d1ae617c2	Returns the weather for a given city and country. Retrieves data from the National Weather Service every 15 minutes. The service is unsecured.

7.5.2 Viewing APIs Subscribed to a Specific Application

To view details for and APIs subscribed to an application, from the Subscriptions page, click the name of the application.

The Application region displays the Name, Key, and Description of the application.

The APIs region lists the APIs to which the selected application is subscribed. The Tags and Description for each API is also displayed.

7.6 Using APIs in Your Applications

This section describes the difference between managed and unmanaged APIs, testing APIs, and using APIs in your applications.

7.6.1 Understanding the Difference Between Managed and Unmanaged APIs

Access to managed APIs is controlled at runtime using an application entitlement key. Calls to the API without this key fail. See [Using Unmanaged APIs](#), for more information.

In order to successfully invoke a managed API, you first need to create an application and subscribe to a given managed API. The client must then invoke the API providing the API Key in the request header for the call to be successful.

Unmanaged APIs can be invoked without subscription and without passing an entitlement key at runtime.

Analytics are available for managed or unmanaged APIs. These details appear at the top of API detail pages, as described in [Data Displayed on API Detail Pages](#).

7.6.2 Using Unmanaged APIs

There are no special actions you need to take to use an unmanaged API. You do not need to subscribe to unmanaged APIs nor do you need to pass an entitlement key at runtime. Invoke unmanaged APIs as you normally would in your application.

7.6.3 Using Managed APIs

You must first subscribe to a managed API before you can use it. See [Subscribing to APIs from the Subscriptions Page](#), for details about subscribing to an API.

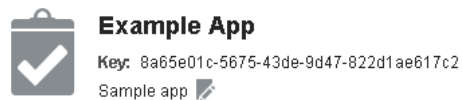
Each application in the API Manager Portal has a unique entitlement key that must be passed at runtime for each API that application is subscribed to. The entitlement key must be passed as an HTTP user header using the X-API-KEY attribute. For example, if an application's entitlement key is b4a4cd59-744c-4a08-a888-6287ce6431c0, the attribute X-API-KEY: b4a4cd59-744c-4a08-a888-6287ce6431c0 must be present in the request header for an invocation of a managed API for the call to be successful.

[Finding an Application's Entitlement Key](#), describes how you find an application's entitlement key in the API Manager Portal.

7.6.4 Finding an Application's Entitlement Key

You find an application's entitlement key on that application's entry on the Subscriptions page.

To find an application's entitlement key, click **Subscriptions**. From the Applications panel, click the name of the application whose entitlement key you want to find. The application's entitlement key is displayed in the Application region, as shown in the following figure.



Note:

You must use this key as described in [Using Managed APIs](#) to invoke a managed API. Invoking a managed API without using the key results in a 403 Forbidden HTTP status code.

Tip:

Hover over an application's key to display a tool tip. You can copy the X-API-KEY attribute containing the application's key directly from this tool tip.

7.6.5 Testing APIs

API producers can choose to supply a URL for testing. If a test URL has been supplied, an if a user with the API Curator role has added that URL to API metadata (described in [Editing the Metadata of an API](#)), API Consumers can test and evaluate the functionality of an API before subscribing to it.

To test an API:

1. Follow the procedure described in [Finding APIs Using the Catalog Page](#), to find an API that you want to test.
2. If available, the testing URL is displayed under the API Details heading in the Testing field.

Tip:

The API Curator is encouraged to include instructions for testing the API either in the Description field of the Overview section or as an HTML reference in the References section. Look in these fields for testing instructions.

3. Use browser plug-ins or other web service testing tools to test the API using the provided URL.

7.7 Unsubscribing from APIs

You can unsubscribe from an API if it is no longer being used. For example, you can unsubscribe from an API if it is deprecated in favor of a newer version or you decide that the API is not a good fit for your application.

After you unsubscribe from a managed API, any invocations to that API using the entitlement key for the application that used to subscribe to it will be rejected.

The two methods you use to unsubscribe to APIs are described in this section.

7.7.1 Unsubscribe from APIs Using the APIs Panel of the Subscriptions Page

To unsubscribe from an API using the APIs panel:

1. Navigate to the Subscriptions page. Ensure that the APIs panel is selected.
2. Find an API in the list from which you want to unsubscribe.
3. Click the **Unsubscribe** icon, as shown in the following figure.



Tip:

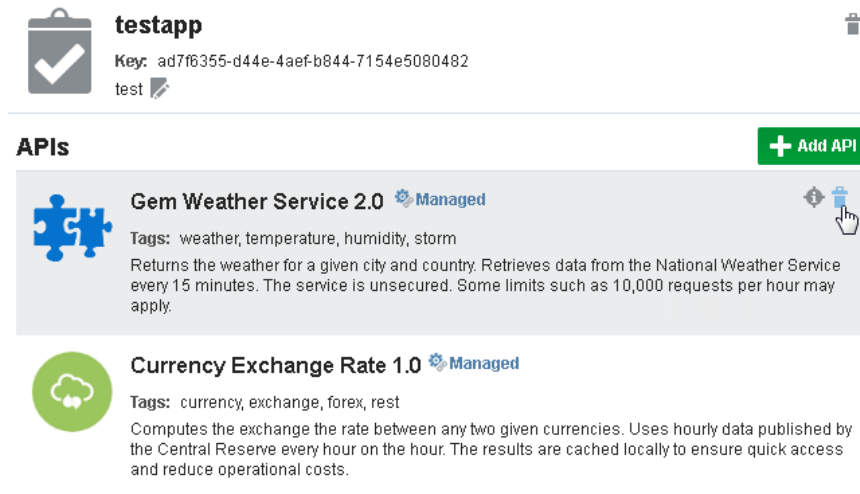
There are multiple entries for an API if more than one application is subscribed to it. Ensure that you click **Unsubscribe** icon for the API entry that corresponds with the application from which you want to remove the subscription.

4. When prompted, click **Yes** to confirm unsubscription.

7.7.2 Unsubscribe from APIs Using the Applications Panel of the Subscriptions Page

To unsubscribe from an API using an Application panel:

1. Navigate to the Subscriptions Page.
2. Click the name of the application from which you want to unsubscribe APIs to display its Application panel.
3. Hover over the API from which you want to unsubscribe and click the **Unsubscribe** icon, as shown in the following figure.



The screenshot displays the Oracle API Manager interface. At the top, there is a header for the application 'testapp' with a key 'ad7f6355-d44e-4aef-b844-7154e5080482' and a 'test' label. Below this is a section titled 'APIs' with a '+ Add API' button. Two APIs are listed:

- Gem Weather Service 2.0** (Managed): Tags: weather, temperature, humidity, storm. Returns the weather for a given city and country. Retrieves data from the National Weather Service every 15 minutes. The service is unsecured. Some limits such as 10,000 requests per hour may apply.
- Currency Exchange Rate 1.0** (Managed): Tags: currency, exchange, forex, rest. Computes the exchange the rate between any two given currencies. Uses hourly data published by the Central Reserve every hour on the hour. The results are cached locally to ensure quick access and reduce operational costs.

4. When prompted, click **Yes** to confirm unsubscription.

7.8 Next Steps

Next steps include administering API Manager.

See [Administering Oracle API Manager](#) for more information.

Administering Oracle API Manager

This chapter describes administration tasks for Oracle API Manager.

The following topics are covered:

- [Viewing Subscription Statistics Using the Subscriptions Page](#)
- [Viewing Analytics in Oracle API Manager](#)
- [Importing and Exporting APIs and API Curation Metadata](#)

8.1 Viewing Subscription Statistics Using the Subscriptions Page

You can view subscription statistics for users and for APIs. User subscription statistics display the applications created for each user and the APIs subscribed to these applications.

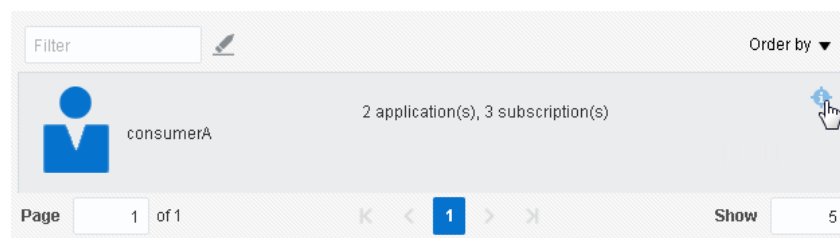
This section describes how to view subscription details. The tasks described here are performed by users with the `APIApplicationAdministrator` role using the API Manager Portal.

8.1.1 Viewing a User's Subscriptions

You can view each user's created applications and the APIs subscribed to these applications.

To view a user's subscriptions:

1. From the Subscriptions page, click **Users**. The Users page displays the number of applications created by and APIs subscribed to each user that has created applications or subscribed to APIs.
2. Click a user's **Details** icon to display that user's Detail page, as shown in the following figure. The User Detail page displays all applications created by the selected user.



3. Click the name of an application to display its details and subscribed APIs.

8.1.2 Viewing API Subscription Details

You can view the users and applications that subscribe to an API.

To view API subscription details:

1. From the Subscriptions page, click **APIs** to display the APIs panel. Each API with subscriptions is displayed. The APIs panel displays the number of users that have subscribed to and the number of subscriptions to each API.
2. Click the **Details** icon for an API to display the Usages tab of an API detail page. This page lists all users and applications that subscribe to this API.

8.1.3 Using the Quick Search Features

Oracle API Manager Portal includes quick search queries that return all applications or APIs that match the query. You can use the quick search to return empty applications, unused APIs (those not subscribed to), and deprecated APIs.

To perform a quick search:

1. From the Subscriptions page, click one of the entries in the Quick Search region: **Empty Applications**, **Unused APIs**, or **Deprecated APIs**.

Tip:

The Deprecated APIs Quick Search returns only deprecated APIs to which a user is subscribed. If a deprecated API has no subscriptions it is returned only by the Unused APIs Quick Search.

2. **(Optional)** You can filter the Quick Search results further by entering search terms into the **Filter** field.

8.2 Viewing Analytics in Oracle API Manager

You can view usage and performance analytics statistics in the API Manager Portal.

The tasks described here are performed by users with the `APIApplicationAdministrator` role.

Note:

You may have to enable API Analytics, as described in [Enabling API Analytics](#), for analytics data to populate in API Manager.

8.2.1 Enabling API Analytics

You must enable the **Monitoring** operational setting for proxy services in Fusion Middleware Control to enable analytics information in API Manager.

Note:

Analytics are available only for services for which the Monitoring operational setting is selected. If you are having difficulty accessing analytics information, ensure that monitoring is enabled for each service for which you want API analytics.

To enable API Analytics:

1. From Fusion Middleware Control, do one of the following to find services:
 - To search for services across the domain, in the Target Navigator expand **SOA** and click **service-bus**.
 - To search for services in a Service Bus project, in the Target Navigator expand **SOA**, expand **service-bus**, and select the name of the project in which to search.
2. Click the **Operations** tab.
3. Specify the search criteria to use to locate the services whose settings you want to modify. You can specify the following criteria, all of which are optional except Type:
 - **Type:** Select **Proxy Services**.
 - **Name:** Enter the name of the services to locate.
 - **Path:** The path (project name and folder names, if any) to the services to locate. If you are on the Service Bus Project Operations tab, the path is already filled in for you.

Tip:

Searching without specifying any of the above criteria will return all services in the domain or project.
4. Click **Search**.
5. For any proxy service in the results list, select the **Monitoring** check box to enable API analytics for the corresponding API, or clear a check box to disable API analytics.

Tip:

To enable or disable API analytics for all the services in the list, select or clear the **Monitoring** check box in the column header.
6. Click **Apply**.

API analytics are now enabled for the selected services.



8.2.2 Viewing Simple Analytics for an API

You can view simple analytics for an API using the Analytics page. A graph representing the total message count and the error message count received over the aggregation interval and the average response time (in milliseconds) is displayed for the API you select.

To view analytics for an API, from the Analytics page, click the **Details** icon for the API for which you want to view simple analytics data, as shown in the following figure.

Analytics

Filter All types Managed and unmanaged

 **Currency Exchange Rate 1.0** Managed 

Tags: forex, currency, exchangerate

Computes the exchange the rate between any two given currencies. Uses hourly data published by the Central Reserve every hour on the hour. The results are cached locally to ensure quick access.

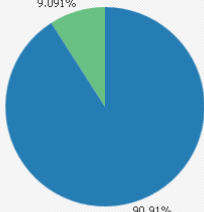
The data is displayed in the pane on the right side of the page, as shown in the following figure.

Analytics

Filter All types Managed and unmanaged

View Statistics Current Aggregation Interval

Past 1 week(s)



770.0 milliseconds
Average Response Time

Page 1 of 2 Show

The analytics displayed are determined by the option selected in the View Statistics list. Select **Current Aggregation Interval** to display analytics for only the current aggregation interval (the Time Period displayed on the detail page), or select **Since Last Reset** to display analytics statistics since the last reset.

8.2.3 Viewing Detailed Analytics for an API

You can view more detailed statistics for an API on its detail page. The data on an API detail page is described in [Analytics Data Displayed on API Detail Pages](#).

To view detailed analytics for an API:

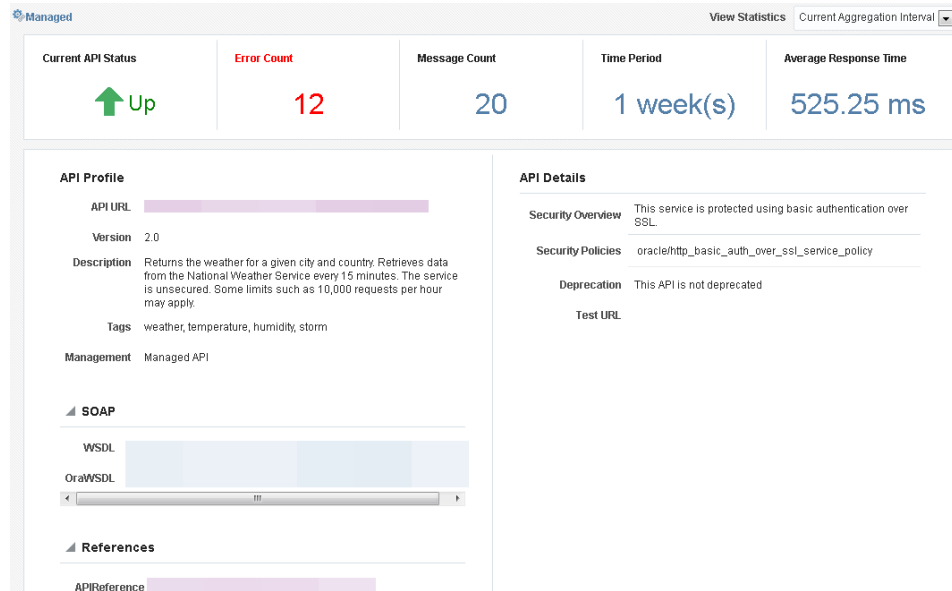
1. From the Catalog page, find the API for which you want to view detailed analytics data.
2. Click the **Details** icon of an API, and then click the **Details** tab to display its details page.
3. **(Optional)** From the View Statistics list, select **Current Aggregation Interval** to display analytics for only the current aggregation interval (the Time Period displayed on the detail page), or select **Since Last Reset** to display analytics statistics since the last reset.

8.2.4 Analytics Data Displayed on API Detail Pages

The API Detail page displays the following status for an API:

- **Managed Status:** Indicates whether the API is managed or unmanaged.
- **Deprecation Status:** Indicates whether the API is deprecated or undeprecated

A sample API Detail page is displayed in the following figure.



The API Detail page displays the following analytics:

- **Current API Status:** Indicates whether the API is up or down.
- **Error Count:** Displays the number of errors received when invoking this API.
- **Message Count:** Displays the number of messages received from the API.
- **Time Period:** Displays the length of time for which the displayed data has been aggregated.
- **Average Response Time:** Displays the average message response time (in milliseconds). See "Configuring Operational and Global Settings" in *Administering Oracle Service Bus* for more information about changing the aggregation interval.

The analytics displayed are determined by the option selected in the View Statistics list. Select **Current Aggregation Interval** to display analytics for only the current aggregation interval (the Time Period displayed on the detail page), or select **Since Last Reset** to display analytics statistics since the last reset.

The API Detail page also displays the following metadata:

Element	Description
API Profile	Displays basic metadata associated with an API.
API URL	Links to the URL of the API.

Element	Description
Version	Displays version information for the API
Description	Contains a detailed description of the functionality of an API.
Tags	Lists tags associated with the API.
Management	Indicates whether the API is managed or unmanaged.
SOAP	Displays links for the associated WSDL file. Note: this region is displayed only for WSDL-based SOAP APIs.
WSDL	Displays a link to the WSDL file.
OraWSDL	Displays a link to the OraWSDL file, if applicable.
REST	Displays a link to the WADL file Note: This region is displayed only for REST APIs.
References	Displays external documentation references and other information sources for the API.
API Details	Displays additional details for the API.
Security Overview	Displays an overview of the Global security policies attached to this API.
Security Policies	Displays the security policies attached to this API. Hover over a policy and click its Details icon to display a policy's description.
Deprecation	Indicates whether the API is deprecated or not deprecated.
Testing	Displays the testing URL for the API.

8.3 Importing and Exporting APIs and API Curation Metadata

When moving from a one environment to another environment, you must export the APIs and the API metadata from the first environment, and then import the data into any additional environments. To help you migrate the data to additional environments, Oracle API Manager exposes Import and Export functions.

This section describes using the import and export functions and provides sample WLST scripts for use in migrating the APIs and API metadata from a test to a production environment. The tasks described here are performed by users with the `Deployer` or `IntegrationAdministrator` roles.

8.3.1 Exporting from an Environment

To export data from an environment:

1. Export a configuration JAR file from the Oracle Service Bus Console containing the required projects and resources by completing the "How to Export Resources to a Configuration JAR File in the Console" task in *Developing Services with Oracle Service Bus*.
2. Ensure that your environment is set up to use WLST and that the necessary libraries are included. The JAR file located at `<MW_HOME>/osb/lib/`

`apimanager-client.jar` must be placed in the classpath for the API Manager import and export calls to function.

See "Getting Started Using the Oracle WebLogic Scripting Tool (WLST)" in *Administering Oracle Fusion Middleware* for more information about using WLST and *Java Reference for API Manager* for API Manager API reference documentation.

3. Create a WLST script to export the curation metadata JAR file from the Service Bus Console. See [Sample Script for Exporting Curation Metadata](#) for a sample WLST export script.
4. Run the script. The curation metadata is exported to a JAR file, the name of and location of which are determined by properties defined in the script.

8.3.2 Sample Script for Exporting Curation Metadata

The following example displays a sample `exportCuration.py` script that can be modified and used for exporting curation metadata from the Service Bus Console.

Example - Sample `exportCuration.py` Script for Exporting Curation Metadata

```
from oracle.apimanager.importexport import CatalogExportPlan

hostname = sys.argv[1]
port = sys.argv[2]
username = sys.argv[3]
password = sys.argv[4]

connect(username, password, 't3://'+hostname+':'+port)

domainRuntime()
cd('DomainServices/AdminService')

print '*****Exporting curation metadata...'
#export curation metadata for all projects
explan=CatalogExportPlan()

filename='<file_path>/<file_name>'

print '*****Exporting curation metadata to' + str(filename)
exportedjar=cmo.exportCatalog(explan)
exportfile=open(filename, 'wb')
exportedjar.toFile(exportfile)
exportfile.close()
print '*****Export curation metadata finished...'

print '*****Exiting...'
exit()
```

8.3.3 Importing into an Environment

To import data into an environment:

1. Create a new session in the Oracle Service Bus Console by completing the "How to Create a Session" task in *Developing Services with Oracle Service Bus*.
2. Import the configuration JAR, created in Step 1 of [Exporting from an Environment](#), into the Service Bus Console by completing the "How to Import Resources from a Configuration JAR File in the Console" task in *Developing Services with Oracle Service Bus*.

3. Activate the Service Bus session by completing the "How to Activate a Session" procedure in *Developing Services with Oracle Service Bus*.
4. Ensure that your environment is set up to use WLST and that the necessary libraries are included. The JAR file located at `<MW_HOME>/osb/lib/apimanager-client.jar` must be placed in the classpath for the API Manager import and export calls to function.

See "Getting Started Using the Oracle WebLogic Scripting Tool (WLST)" in *Administering Oracle Fusion Middleware* for more information about using WLST and *Java Reference for API Manager* for API Manager API reference documentation.
5. Create a WLST script to import the curation metadata JAR file, created in Step 3 of [Exporting from an Environment](#), into the Service Bus Console. See [Sample Script for Importing Curation Metadata](#) for a sample WLST import script.
6. Run the script. The metadata from the JAR file is imported into the Service Bus Console. API metadata appears in the API Manager Portal when an API is published.

8.3.4 Sample Script for Importing Curation Metadata

The following example displays a sample `importCuration.py` script that can be modified and used for importing curation metadata into the Service Bus Console.

Example - Sample `importCuration.py` Script for Importing Curation Metadata

```
from oracle.apimanager.importexport import CatalogImportPlan
from java.util import HashSet

hostname = sys.argv[1]
port = sys.argv[2]
username = sys.argv[3]
password = sys.argv[4]

connect(username, password, 't3://'+hostname+':'+port)

domainRuntime()
cd('DomainServices/AdminService')

print 'Importing curation metadata...'
implan=CatalogImportPlan()

#select projects to be imported
#if project set is empty, all projects will be included
projects=HashSet()
projects.add('SOAPProject')
implan.setProjects(projects)

print 'import plan: '
getproj=implan.getProjects()
for gp in getproj:
    print str(gp)

jar=open('<file_path>/<file_name>', "rb").read()

result=cmo.importCatalog(jar,implan)
jar.close()
print 'Import curation metadata finished...'
```



```
print '*****Printing All entries in archive...'  
content=result.getContent()  
for ct in content:  
    print ct.getFullName()  
  
print '*****Printing imported entries...'  
importset=result.getImported()  
for ref in importset:  
    print ref.getFullName()  
  
print '*****Printing skipped entries...'  
skipset=result.getSkipped()  
for skref in skipset:  
    print skref.getFullName()  
  
print '*****Exiting...'  
exit()
```

