Oracle® Cloud

Using Oracle API Platform Cloud Service - Classic

E96996-02
January 2019

Oracle Cloud Using Oracle API Platform Cloud Service - Classic,

E96996-02

Primary Author: Oracle Corporation

Contents

3 Managing Gateways

4 Manage APIs

7 Managing Applications

8 Use Analytics

9 Frequently Asked Questions for Oracle API Platform Cloud Service - Classic

Preface

Topics:

Using Oracle API Platform Cloud Service describes how to manage gateways and APIs, deploy APIs to gateways, and publish APIs to the Developer Portal with Oracle Oracle API Platform Cloud Service.

Audience

Using Oracle API Platform Cloud Service is intended for Administrators, API Managers, and Gateway Managers who want to manage, secure, document, and publish APIs and manage gateways with Oracle API Platform Cloud Service.

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.

Related Resources

See these Oracle resources:

- Oracle Public Cloud

  `http://cloud.oracle.com`
- *Consuming APIs with the Oracle API Platform Cloud Service Developer Portal*
- *REST API for the Administration Service in Oracle API Platform Cloud Service*
- *REST API for the Analytics Service in Oracle API Platform Cloud Service*
- *REST API for the Consumer Service in Oracle API Platform Cloud Service*

- *REST API for the Gateway Controller in Oracle API Platform Cloud Service*
- *REST API for LifeCycle Management in Oracle API Platform Cloud Service*
- *REST API for the Management Service in Oracle API Platform Cloud Service*

Conventions

The following text conventions are used in this document:

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

1

Getting Started with Oracle API Platform Cloud Service - Classic

Review the following topics to learn about how Oracle API Platform Cloud Service- Classic works. These topics provide information about Oracle API Platform Cloud Service- Classic concepts and components to help you get started managing your APIs.

Topics

Learn About Oracle API Platform Cloud Service - Classic

Create, manage, secure, and advertise APIs to connect to new or existing services.

An API that's hosted on Oracle API Platform Cloud Service - Classic has the following features:

- It enables authorized and authenticated developers of mobile and web apps to access and consume your organization’s services.
- It can be deployed on-premises or to any cloud service including Oracle Cloud, Amazon Web Services Cloud, and Microsoft Azure Cloud, thus allowing the API and your services to reside in the same place.
- It enables access to legacy applications and services without modifying the legacy code base.
- It can route requests to more than one service.

Learn About the Components of Oracle API Platform Cloud Service - Classic

Oracle API Platform Cloud Service - Classic includes the My Services Console, the Gateway, the Management Portal, and the Developer Portal components.

- **My Services Console**: Used to provision new service instances, start and stop service instances, initiate backups, and perform other lifecycle management tasks.
- **Gateway**: The security and access control runtime layer for APIs. Each API is deployed to a gateway node from the Management Portal or via the REST API.
- **Management Portal**: Used to create and manage APIs, deploy APIs to gateways, and manage gateways, and create and manage applications. You can also manage and Deploy APIs and manage gateways with the REST API.
- **Developer Portal**: Used by application developers to subscribe to APIs and get the necessary information to invoke them from this portal.

Before You Begin with Oracle Oracle API Platform Cloud Service - Classic

Before you begin using Oracle API Platform Cloud Service - Classic, you must have subscriptions to Oracle Identity Cloud Service, Oracle Database Cloud Service (not Database Schema) and Oracle Storage Cloud Service, and you must have created specific details in each service. You are also required to have a secure shell (SSH) public/private key pair. If you want to host gateway nodes in Oracle's cloud, you must also have a subscription to Oracle Cloud Infrastructure Compute Classic. You may also want to consider a subscription to a higher tier of Oracle Cloud Infrastructure Compute Classic.

- **Oracle Cloud Infrastructure Compute Classic**

  Oracle API Platform Cloud Service - Classic uses Oracle Identity Cloud Service for identity management. The Foundation tier is included. This tier includes basic user and group management features, but lacks the following features:

  - Identity Synchronization
  - User Self-Registration
  - Self-Service Password Reset
  - Self-Service Access Request
  - SSO for Third-Party Cloud Apps
  - Multi-Factor Authentication (MFA)
  - External Identity Provider Federation
  - Social Authentication
  - User Provisioning and Synchronization for Third-Party Cloud Apps
  - Oracle Identity Manager Connector for Oracle Identity Cloud Service
  - Enterprise SLA (99.95%)

  Depending on your use case, you may require a license to one of the other pricing tiers. For example, users managed in an Oracle Identity Cloud Service Basic tier identity domain typically correspond to application developers outside of your company who subscribe to APIs you manage and publish. Users managed in an Oracle Identity Cloud Service Standard tier identity domain typically correspond to API Managers, Gateway Managers, other users who manage resources with Oracle API Platform Cloud Service - Classic, and internal application developers.

  See About Oracle Identity Cloud Service Pricing Tiers and Features.

- **Oracle Database Cloud Service**

  Oracle API Platform Cloud Service - Classic uses Oracle Database Cloud Service to host the database schemas the service requires.

  In your Oracle Database Cloud Service, you need to create a service instance using the standard service level.

  Note the following before creating a service instance in Oracle Database Cloud Service:

  – Do not use the Virtual Image service level for Oracle Database Cloud Service as it does not work correctly during backup and restore.

  See Getting Started with Database Cloud Service in *Administering Oracle Database Cloud Service*.

  When you are ready to create an Oracle API Platform Cloud Service - Classic instance, you are prompted for the following information about your service instance in Oracle Database Cloud Service:

  – Name of the service instance that is up and running in Oracle Database Cloud Service

  – Pluggable database name (for Oracle Database 12c only)

  – Database administrator user name and password

- **Oracle Storage Cloud Service**

  Older backups in Oracle API Platform Cloud Service - Classic are automatically moved to a container in Oracle Storage Cloud Service. Before you can create containers and objects, you must have an active subscription to Oracle Storage Cloud Service. See How to Begin with Oracle Storage Cloud Service Subscriptions in *Using Oracle Cloud Infrastructure Object Storage Classic* .

  See the tutorial Creating Oracle Storage Cloud Service Containers Using the REST API.

  When you're ready to create an Oracle API Platform Cloud Service - Classic instance, you are prompted for the following information about a storage container:

  – Name of the container in Oracle Storage Cloud Service

  – Oracle Storage Cloud Service user name and password

- **Virtual Machine (VM) public key**

  An SSH public/private key pair is used for authenticating access to a VM through an SSH client. You generate an SSH public/private key pair using a standard SSH key generation tool. See Creating SSH Keys for Use with Oracle Cloud Services.

  You may use the same SSH public/private key pair that you used for creating a service instance in Oracle Database Cloud Service.

  When you're ready to create an Oracle API Platform Cloud Service - Classic instance, you're prompted to supply the VM public key. You can also create a new key during the provisioning process and use it to access the VM. See Creating an Oracle API Platform Cloud Service - Classic Instance.

  To connect to a VM in an Oracle API Platform Cloud Service - Classic instance, you'd supply the paired private key when logging in to the machine using an SSH client.

Prior to using Oracle API Platform Cloud Service - Classic, ensure also you're familiar with the following:

- **Oracle Cloud**

  Create and configure your account on Oracle Cloud. See Purchasing a Subscription to Oracle Public Cloud Services.

- **Oracle Compute VMs**

  Oracle API Platform Cloud Service - Classic runs on Oracle Compute VMs. See Getting Started with Oracle Compute Cloud Service to learn about disk images, compute shapes, storage volumes, public IP addresses, network groups, access rules, and SSH public/private key pairs.

- **Oracle WebLogic Server**

  Gateway nodes run on Oracle WebLogic Server 12.1.3 instances.

Prerequisites to Provisioning Oracle API Platform Cloud Service - Classic

Review this section to learn about some things you have to do before you can provision Oracle API Platform Cloud Service - Classic instances.

Selecting an Oracle Storage Cloud Service When Provisioning an Oracle Database Cloud Service

When provisioning an Oracle Database with the Oracle Database Backup Cloud Service provisioning wizard, you must select an Oracle Storage Cloud Service even though this field is optional. If you do not select an Oracle Storage Cloud Service, when you run the Oracle API Platform Cloud Service - Classic provisioning wizard and select this Oracle Database Backup Cloud Service, instance provisioning fails. Always select an Oracle Database Backup Cloud Service that has an Oracle Storage Cloud Service associated with it.

When you configure the backup destination be sure to select **Both Cloud Storage and Local Storage** from the dropdown menu. If you don't select **Both Cloud Storage and Local Storage**, theOracle API Platform Cloud Service - Classic provisioning wizard will not present that Oracle Database Backup Cloud Service instance as an available database in the Oracle API Platform Cloud Service - Classic provisioning wizard.

Selecting an Oracle Database Cloud Service When Provisioning an Oracle API Platform Cloud Service - Classic

Oracle API Platform Cloud Service - Classic uses Database Cloud Service to host the Oracle Fusion Middleware component schemas required by Oracle Java Required Files (JRF). Make sure you have a subscription to Database Cloud Service (Database as a Service), NOT Database Cloud Service (Database Schema).

> **Note:**
>
> To work correctly with Oracle API Platform Cloud Service - Classic, the Database Cloud Service must have backup enabled and be set up to use **Both Local and Cloud Storage**.

How to Begin with Oracle API Platform Cloud Service - Classic Subscriptions

Here's how to get started with Oracle API Platform Cloud Service - Classic paid subscriptions:

1. Purchase a subscription. See Buying a Metered Subscription to an Oracle Cloud Service in *Getting Started with Oracle Cloud*.
2. Set up your Oracle Public Cloud Services account. See Setting Up Oracle Public Cloud Services Account in *Getting Started with Oracle Cloud*.
3. Ensure that you have met all of the prerequisites and subscribed to the required Oracle Cloud Services. See Before You Begin with Oracle Oracle API Platform Cloud Service - Classic.
4. Provision your Oracle API Platform Cloud Service - Classic instance. See Creating an Oracle API Platform Cloud Service - Classic Instance.
5. Verify Oracle API Platform Cloud Service - Classic is ready to use. See Verifying Oracle Public Cloud Services Are Running in *Getting Started with Oracle Cloud*.
6. Learn about the roles. See Roles.
7. Create accounts for your users and assign them appropriate privileges and roles. See Managing Users and Groups and Managing Roles.
8. Install at least one gateway node. See Installing a Gateway Node.

Accessing the My Services Console

To access the My Services console:

1. Display the Sign In to Oracle Cloud page by clicking the **My Services** URL link in your Welcome email or by following these instructions:
   a. Open your web browser and go to the Oracle Cloud website: `http://cloud.oracle.com`.
   b. Click **Sign In**.
   c. In the **Cloud Account** list, select the data center where your services are located, either **Public Cloud Services - US** or **Public Cloud Services - EMEA**, and then click the **My Services >** button.
2. On the Sign In to Oracle Cloud page, enter your identity domain, and then, click **Go**.

   > **Note:**
   >
   > If asked to provide a service type, enter `APICS`. This is Oracle API Platform Cloud Service - Classic service type.

3. Enter your username and password, and then click **Sign In**.

   The My Services dashboard opens.

4. Click the  navigation menu in the My Services dashboard and then click **API Platform Cloud Service**.
5. If a Welcome page is displayed, view the list of instances by clicking the **Services** tab.

Creating an Oracle API Platform Cloud Service - Classic Instance

Create an Oracle API Platform Cloud Service - Classic instance with the Provisioning wizard in the My Services Console.

Before You Begin

Ensure that you have subscribed to the required services, collected the required information for each service, and created SSH keys. See Before You Begin with Oracle Oracle API Platform Cloud Service - Classic.

To create a new Oracle API Platform Cloud Service - Classic instance:

1. Open the My Services Console. See Accessing the My Services Console.
2. Click **Create Service**.

   If no DBaaS service is available, a message displays and the service creation halts until there is a running DBaaS service is available.
3. On the Service page, provide a name and description for the service instance, and provide information about its high-level characteristics. When you are finished, click **Next** to advance to the Service Details page.

Element	Description
Service Name	Use the following conventions to compose your instance name: • The name must start with a letter. • The name cannot contain more than 50 characters. • The name cannot contain special characters other than the hyphen character. A message displays if your instance name does not meet all criteria.
Description	Describe the instance.
Notification Email	**(Optional)** Enter an email address where you would like updates about the instance-creation operation to be sent.

Element	Description
SSH Public Key	Specify the value of the VM Public Key, the name of the file that contains the public key value, or have the wizard create a new key for you. Define the public key for the secure shell (SSH). This key is used for authentication when connecting to the Oracle API Platform Cloud Service - Classic instance using an SSH client. Click **Edit** to display the public key input for VM access and specify the public key using one of the following methods: • Select **Key file name** and click **Choose File** to select a file that contains the public key for the secure shell (SSH). • Select **Key value** and enter the value of an existing public key. • Select **Create a New Key** to have the wizard create a new public key for you. Make sure you download the zip file containing the keys that the wizard generated. Click **Enter** when you're finished adding or generating the SSH key.

4. Complete the **Database Configuration** section to provide details about the Database Cloud Service instance you've created for this Oracle API Platform Cloud Service - Classic instance.

Element	Description
Associated DBaaS Service Name	Select an existing Database Cloud Service instance name.
PDB Name	Enter the name of the pluggable database for Oracle Database 12c. If not specified, the PDB name provided when the Oracle Database Cloud Service database deployment was created will be used. This value does not apply to Oracle Database Cloud Service database deployments running Oracle Database 11g.
Administrator User Name	Your Database Cloud Service user name. This value must be set to a database user with SYSDBA system privileges. You can use the default user SYS or any user that has been granted the SYSDBA privilege.
Password	The database administrator password specified when the Database Cloud Service instance was created.

5. Complete the **Load Balancer Configuration** section to provide details about the load balancer provisioned with your instance.

Element	Description
Provision Load Balancer	Oracle API Platform Cloud Service - Classic automatically provisions a load balancer. This option is set to **true** by default and is not editable.

6. Complete the **API Gateway** section to provide details about the number of gateway licenses you want this instance to use.

Element	Description
Gateway Licenses	The number of gateway licenses you want this instance to use.

7. Complete the **Weblogic** section to provide details about the WebLogic administrator account for your instance.

Element	Description
Administration User Name	The user name of the Oracle WebLogic Server Administrator. The Administrator User Name for the Weblogic server should start with a letter, consist of letters and numbers, and be 8 to 50 characters. A message displays if your user name does not meet all criteria. Note that you can change the user name through the WebLogic Server Administration Console after you have created the instance.
Administration User Password	Specify an Oracle WebLogic Server Administrator password that meets the following criteria: • It must begin with a letter. • It must contain between 8 and 30 characters. • It must contain at least one number. • Optionally, it can contain any number of the following special characters: “$#_”. For example: `Ach1z0#d`. A message displays if your password does not meet all criteria.
Confirm Administration User Password	Retype the password.

8. Complete the **Backup and Recovery Configuration** section, providing information about the Oracle Storage Cloud Service container where cloud backups are stored.

Element	Description
Cloud Storage Container	The URL of the Object Storage Classic container in which backups must be stored **Format**: `[http|https]://storage_service_rest_endpoint_url/{containerName}` **Example**: `https://example.storage.oraclecloud.com/v1/MyService-example/MyContainer` You can find the REST endpoint URL on the Service Details page in the My Services Console. See Finding the REST Endpoint URL for Your Service Instance in *Using Oracle Storage Cloud Service*. By default, all backups that are more than seven days old are moved to the storage service at this URL.
Cloud Storage User Name	The name of the Oracle Storage Cloud Service administrator. The Cloud Storage user name should be between 2 to 50 characters and consist of only letters, numbers and the following special characters: .(dot), -(hyphen), _(underscore), @(at). A message displays if your user name does not meet all criteria.
Cloud Storage Password	The password of the Oracle Storage Cloud Service administrator. Cloud Storage Password should be a minimum of 8 and maximum 30 characters. A message displays if your password does not meet all criteria.

9. Click **Next** to advance to the Confirmation page.
10. On the Confirmation page, review the information listed. If you are satisfied with the information, click **Create**.

    If you need to change the information, use the navigation bar or **Previous** button at the top of the wizard to step back through the pages in the wizard.
    Click **Cancel** to cancel out of the wizard without creating a new instance.

> **Important:**
>
> After you have provisioned a service instance, you need to create users and install and register at least one gateway node before you begin using the service. See Managing Users and Groups and Installing a Gateway Node.

Accessing the Oracle API Platform Cloud Service - Classic Management and Developer Portals

To access the Oracle API Platform Cloud Service - Classic Management and Developer Portals:

> **Note:**
>
> This task describes accessing the Oracle API Platform Cloud Service - Classic Management and Developer Portals from the My Services Console. You can also access the Management and Developer Portals directly using these URL formats:
>
> - `https://<OTD_IP>/apiplatform` for the Management Portal
> - `https://<OTD_IP>/developers` for the Developer Portal
>
> where *<OTD_IP>* is the IP address of the Oracle Traffic Director load balancer associated with your service instance.

1. Sign in to the My Services Console. See Accessing the My Services Console.
2. On the Services page, click the  Manage this service icon for your instance, and then click **Open API Portal Console** to open the Management Portal, or click **Open Developer Portal Console** to open the Developer Portal.
3. When the login page appears, enter a username and password of a user assigned the proper role for the instance you're accessing (for example, sign in to the Management Portal with a user assigned the Administrator, API Manager, or Gateway Manager role and sign in to the Developer Portal with a user assigned the Administrator, API Manager, or Application Developer role).

About Oracle API Platform Cloud Service - Classic Roles, Resources, Actions, and Grants

Learn about roles, resources, actions, and grants in Oracle API Platform Cloud Service - Classic.

Topics

Terms Used by User Management

These terms are used throughout Oracle API Platform Cloud Service - Classic to define user management concepts.

Entity	Description
User	An Oracle API Platform Cloud Service - Classic user. Users can be members of groups or roles.
Group	A group of users. Groups can be members of other groups or roles.
Role	A role is a group which is predefined by the system Roles. It cannot be a member of another role or group. The ability to perform a certain action is determined by membership in a role and optionally a grant on the resource(s) being acted upon.
Action	A fine-grained operation by the user; for example, CreateApplication, DeleteAPI, etc.
Grant (noun)	A permission to perform a set of actions on a specific resource. Grants apply to one resource type. For example, the DeployToGatewayGrant can only be applied to Gateway type resources. Grants are granted to a user or group
Resource	An object being acted on by a user; for example, a single gateway or API, as opposed to all gateways or APIs. Resources have a resource type.
Resource Type	The type of resource, like API, gateway, or application.

Roles

Roles determine which interfaces a user is authorized to access and the grants they are eligible to receive. You can assign one or more of these roles to Oracle API Platform Cloud Service - Classic users and groups: Administrator, API Manager, Application Developer, Gateway Manager, Gateway Runtime, Service Manager, and Plan Manager.

The table below describes each of the available roles.Managing Roles describes how you assign roles.

Name	Description
Administrator	System Administrators responsible for managing the platform settings. Administrators possess the rights of all other roles and are eligible to receive grants for all objects in the system. Administrator tasks are described in Administering Oracle API Platform Cloud Service - Classic.
API Manager	People responsible for managing the API lifecycle, which includes designing, implementing, and versioning APIs. Also responsible for managing grants and applications, providing API documentation, and monitoring API performance. API Manager tasks are described in Manage APIs.

Name	Description
Application Developer	API consumers granted self-service access rights to discover and register APIs, view API documentation, and manage applications using the Developer Portal. Application Developer tasks are described in Getting Started with the API Platform Cloud Service Developer Portal in *Consuming APIs with the Oracle API Platform Cloud Service - Classic Developer Portal*.
Gateway Manager	Operations team members responsible for deploying, registering, and managing gateways. May also manage API deployments to their gateways when issued the Deploy API grant by an API Manager. Gateway Manager tasks are described in Managing Gateways.
Gateway Runtime	This role indicates a service account used to communicate from the gateway to the portal. This role is used exclusively for gateway nodes to communicate with the management service; users assigned this role can't sign into the Management Portal or the Developer Portal.
Service Manager	People responsible for managing resources that define backend services. This includes managing service accounts and services. Service Manager tasks are described in Managing Services and Service Accounts.
Plan Manager	People responsible for managing plans. Plan Manager tasks are described in Managing Plans.

Resource Types

You issue grants for individual resources in Oracle API Platform Cloud Service - Classic. This gives you fine-grained control over which users can perform which actions on a resource. You can issue grants for APIs, applications, gateways, and plans.

Administrators can issue grants to all users for all resources. Users with a role associated with a resource type and the Manage grant for a resource can issue grants for that resource. For example, Gateway Managers with the Manage Gateway grant for a specific gateway can issue grants for it. Gateway Managers without the Manage Gateway grant for a gateway can't issue grants for it.

Resource Type	Description
API	An API that is managed in Oracle API Platform Cloud Service - Classic.
Application	An external application that is registered to an API/plan.
Gateway	A gateway, managed in Oracle API Platform Cloud Service - Classic, that you deploy APIs to. The gateway runtime acts as the security layer, enforcing policies applied to APIs and routing requests to backend services. You issue grants to the logical gateway, not individual gateway nodes. Grants issued to the logical gateway apply to all nodes registered to it.

Resource Type	Description
Plan	A plan is a set of APIs and specific policies for those APIs.
Service Account	A Service Account provides authentication configuration for outbound calls. You define a service account resource once and reuse it in policies where this account is required to access services.
Service	A Service provides configuration and access to a backend service. You define a service resource once and reuse it any number of policies.

Actions

Grants determine the actions users can perform on a resource.

Action	Resource	Display Name	Description
APICreate	GenericResource	Create API	Create an API
APIDelete	API	Delete	Delete an API
APIDeploy	API	Deploy	Deploy or request deployment for this API to a gateway. The user also needs the appropriate permission on the Gateway resource
APIEditAll	API	Edit	Modify the API
APIEditPublic	API	Edit Public Properties	Modify the Public details of an API (e.g. a doc person)
APIGrantDeployAPI	API	Grant Deploy API	Give a gateway manager permission to deploy this API (issue the DeployAPIGrant grant)
APIGrantManageAPI	API	Grant Manage API	Give another APIManager the permission to manage this API (issue the ManageAPI grant)
APIGrantViewAllDetails	API	Grant View All Details	Give another user permission to view the API's (full) details. (issue the ViewAllDetailsAPIGrant grant)

Action	Resource	Display Name	Description
APIModifyLifecycleState	API	Grant View Public Details	Give another user permission to view the API's public details in the Developer Portal (issue the ViewPublicDetailsAPI Grant grant)
APIModifyPublishState	API	Modify Lifecycle State	Changes the lifecycle state of the API
APIResume	API	Modify Publish State	Publish the API to the developer portal or remove it from the portal
APISuspend	API	Resume	Resume a deployed API on a gateway
APIUndeploy	API	Suspend	Suspend a deployed API on a gateway
APIViewAllDetails	API	Undeploy	Undeploy this API to a gateway. The user also needs the appropriate permission on the Gateway resource
APIViewHistory	API	View All Details	View all data about the API
APIViewPublicDetails	API	View Deployment Details	View data needed for managing the API deployment
ApplicationCreate	API	View History	View the history of updates made to the API
ApplicationDelete	API	View Public Details	View data meant for external consumption (primarily for Developer Portal use)
ApplicationEditAll	GenericResource	Create Application	Create a new Application
ApplicationEditByManager	Application	Delete	Delete this Application
ApplicationGrantManageApplication	Application	Edit	Modify the properties of an application
ApplicationGrantViewAllDetails	Application	Edit a subscribeed application	Allows the API Manager to edit a subset of properties of an application subscribeed to an API

Action	Resource	Display Name	Description
ApplicationIssueKey	Application	Grant Manage Application	Give someone else the ManageApplicationGrant so they can modify this application (issue the ManageApplicationGrant grant)
ApplicationSubscribe	Application	Grant View All Details	Give someone the ViewAllDetailsApplicationGrant
ApplicationRegistrationResume	Application	Issue an Application Key	Issues a new application key
ApplicationRegistrationSuspend	Application	Subscribe	Subscribe or request an application registration to an API
ApplicationUnsubscribe	Application	Resume	Resume an application
ApplicationViewAllDetails	Application	Suspend	Suspend an application
ApplicationViewHistory	Application	Unsubscribe	Unsubscribe an application from an API
ApplicationViewAllDetails	Application	View All Details	View the properties of an Application and analytics
ApplicationViewHistory	Application	View History	View the history of updates made to the Application
ApplicationViewManagerDetails	Application	View as API Manager	View the properties needed as an API Manager or Gateway Manager
DeveloperPortalLogin	GenericResource	Developer Portal Login	Login to the ApplicationDeveloper Portal
GatewayApproveDeployRequest	Gateway	Approve API Deployment Request	Approve another users request to deploy and API to this gateway.
GatewayCreate	GenericResource	Create Gateway	Create a new Gateway
GatewayDelete	Gateway	Delete Gateway	Delete a Gateway
GatewayDeploy	Gateway	Deploy an API	Deploy an API to this Gateway
GatewayEditAll	Gateway	Edit All	Modify the gateway properties

Action	Resource	Display Name	Description
GatewayGrantDeploy	Gateway	Grant Deploy	Give another user the ability to deploy APIs to this gateway. (issue the DeployAPIToGateway Grant grant)
GatewayGrantManageGateway	Gateway	Grant Manage Gateway	Give another Gateway Manager the right to manage this gateway. (issue a ManageGatewayGrant grant)
GatewayGrantRequestDeployAPI	Gateway	Grant Request Deploy	Give another user the ability to request a deployment of APIs to this gateway. (issue the RequestDeployAPIToGatewayGrant grant)
GatewayGrantServiceGateway	Gateway	Grant Service Gateway	Give a service account the ability to retrieve configurations and post statistics from this gateway
GatewayGrantViewGateway	Gateway	Grant View All Details	Give another user the ability to view Gateway details (issue the ViewGatewayGrant)
GatewayRequestDeploy	Gateway	Request Deployment of an API	Request an API be deployed to this Gateway. Someone with GatewayDeploy needs to do the actual Deploy
GatewayRetrieveConfiguration	Gateway	Retrieve Configuration	Retrieve gateway configuration updates from the portal. (Used by GatewayRuntime service accounts only)
GatewayUndeploy	Gateway	Undeploy an API	Undeploy an API from this Gateway
GatewayUploadStatistics	Gateway	Upload Statistics	Upload gateway runtime statistics to portal. (Used by GatewayRuntime service accounts only)
GatewayViewAllDetails	Gateway	View All Details	View all data about the gateway
GatewayHistoryView	Gateway	View History	View the history of updates made to the Gateeway

Action	Resource	Display Name	Description
ManagerPortalLogin	GenericResource	Manager Portal Login	Login to the Management Portal
PlanApproveRegistration	Plan	Approve Application Registration	Approve a request to subscribe and application to use a Plan
PlanCreate	GenericResource	Create Plan	Create a new Plan
PlanDelete	Plan	Delete	Delete the plan
PlanEditAll	Plan	Edit	Edit all properties of the plan
PlanEditPublic	Plan	Edit Public Details	Edit the public properties of the plan.
PlanGrantManagePlan	Plan	Grant Manage Plan	Give another API Manager the ability to manage this plan (issue the ManagePlanGrant)
PlanGrantSubscribeApplication	Plan	Grant Subscribe	Give an Application Developer the ability to subscribe an application for this plan (issue the SubscribeApplicationForPlanGrant grant)
PlanGrantRequestSubscribeApplication	Plan	Grant Request Subscription	Give an Application Developer the ability to request an application be subscribed for this plan (issue the RequestSubscribeApplicationForPlanGrant)
PlanGrantViewAllDetails	Plan	Grant View All Details	Give another user the ability to view all properties of the plan
PlanGrantViewPublicDetails	Plan	Grant Public Details	Give another user the ability to view the plan in the developer portal (issue the ViewPublicDetailsforPlanGrant grant)
PlanGrantEntitleAPI	Plan	Grant Entitle API	Give an API Manager the ability to entitle an API to this plan (issue the EntitleAPIToPlanGrant grant)
PlanModifyPublishState	Plan	Modify Publish State	Modify the publish state of the plan
PlanModifyState	Plan	Modify State	Modify the state of the plan

Action	Resource	Display Name	Description
PlanEntitleAPI	Plan	Entitle API	Entitle an API to an Plan.
PlanSubscribeApplication	Plan	Subscribe Application	Subscribe an Application to have access to an API. No approval needed.
PlanRequestSubscribeApplication	Plan	Request Application Subscription	Request an application be subscribed for use
PlanViewAllDetails	Plan	View All Details	View all details of the plan
PlanViewHistory	Plan	View History	View the history of updates made to the Plan
PlanViewPublicDetails	Plan	View Public Details	View information available to Application Developers in the Developer Portal. **Note**: this action also implies the permission to view the public details of any API which is part of the plan.
PlanEntitleAPI	Plan	Entitle	Entitle an API to an Plan.
PolicyManage	GenericResource	Manage Policies	Upload or update a custom policy
ServiceAccountEditAll	Service Account	Edit	Edit all properties of the service account
ServiceAccountViewAllDetails	Service Account	View All Details	View all details of the service account
ServiceAccountViewHistory	Service Account	View History	View the history of updates made to the service account
ServiceAccountDelete	Service Account	Delete	Delete the service account
ServiceAccountReference	Service Account	Reference	Reference the service account
ServiceAccountGrantManageServiceAccount	Service Account	Grant Manage Service Account	Give another Service Manager the ability to manage the service account
ServiceAccountGrantViewAllDetails	Service Account	Grant View All Details	Give another user the ability to view all properties of the service account

Action	Resource	Display Name	Description
ServiceAccountGrantReferenceServiceAccount	Service Account	Grant Reference Service Account	Give another user the ability to reference a service account
ServiceEditAll	Service	Edit	Edit all properties of the service
ServiceModifyState	Service	Modify State	Edit the state of the service
ServiceViewAllDetails	Service	View All Details	View all details of the service
ServiceViewHistory	Service	View History	View the history of updates made to the service
ServiceDelete	Service	Delete	Delete the service
ServiceReference	Service	Reference	Reference the service
ServiceGrantManageService	Service	Grant Manage Service	Give another Service Manager the ability to manage the service
ServiceGrantViewAllDetails	Service	Grant View All Details	Give another user the ability to view all properties of the service
ServiceGrantReferenceService	Service	Grant Reference Service	Give another user the ability to reference the service
UIPlatformSettingsTab	GenericResource	View Platform Settings Tab	Display the Platform settings tab in API Manager Portal, where Administrator can set tenant level settings (Eg, Time zone)
UIViewAPITab	GenericResource	View API Tab	Display the API tab in Manager Portal
UIViewApplicationTab	GenericResource	View Application Tab	Display the Application tab in Manager Portal
UIViewGatewayTab	GenericResource	View Gateway Tab	Display the Gateway tab in Manager Portal
UIViewRoleTab	GenericResource	View Role Tab	Display the Role tab in Manager Portal
UsersManage	GenericResource	Manage Users	Modify Users, groups, and membership for groups and roles.
UsersViewHistory	GenericResource	View user management history	View change history for users, groups, and roles

Action	Resource	Display Name	Description
ViewAllHistory	GenericResource	View all history across the system	View the change history for all resources and system changes

Grants

In tandem with roles, grants determine which users can access which resources in Oracle API Platform Cloud Service - Classic.

Roles determine which grants a user is eligible to receive; grants determine which actions a user can perform on specific resources. Because grants are issued at the resource level, you have fine-grained control over which users can perform which actions on specific resources. You can control how you want to manage the API lifecycle by issuing certain grant combinations to your users. For example, if you want trusted API Managers to be able to deploy directly to gateways in a development environment without explicit approval from a Gateway Manager, an Administrator or a Gateway Manager can issue that user the Deploy to Gateway grant for a development gateway. In this example the API Manager has not been given approval to deploy directly to a production gateway. They are not able to deploy APIs to it unless they are given explicit approval to do so.

Oracle API Platform Cloud Service - Classic grants, the users each grant can be issued to, and the actions each grant enables are described below.

> **Note:**
>
> Administrators possess the rights of all other roles and are eligible to receive grants for all objects in the system.

API Grants

Grant Name	Description	Can Be Issued To	Associated Actions
Manage API	People issued this grant are allowed to modify the definition of and issue grants for this API.	API Managers	APIDelete APIViewAllDetails APIViewPublicDetails APIEdit APIEditPublic APIModifyPublishState APIModifyLifecycleState APIDeploy APIGrantManageAPI APIGrantViewAllDetails APIGrantViewPublicDetails APIGrantDeployAPI

Grant Name	Description	Can Be Issued To	Associated Actions
View all details	People issued this grant are allowed to view all information about this API in the Management Portal.	API Managers, Gateway Managers, Plan Managers	APIViewAllDetails
View public details	People issued this grant are allowed to view the publicly available details of this API on the Developer Portal. This grant can be issued to users of any role.	API Managers, Application Developers, Plan Managers	APIViewPublicDetails
Entitle API	Users issued this grant are allowed to entitle this API to a plan for which they have entitle rights.	API Managers, Plan Managers	APIEntitlementAdd APIEntitlementEdit APIEntitlementRemove APIEntitlementModifyState APIEntitlementModifyPublishState
Deploy API	API Managers with the Manage API grant already have this permission for all gateways they are allowed to view. API Managers without the Manage API grant and Gateway Managers issued this grant are allowed to deploy or undeploy this API to a gateway for which they have deploy rights. This allows Gateway Managers to deploy this API without first receiving a request from an API Manager.	API Managers, Gateway Managers	APIDeploy

Gateway Grants

Grant Name	Description	Can be Issued To	Associated Actions
Manage Gateway	People issued this grant are allowed to manage API deployments to this gateway and manage the gateway itself.	Gateway Managers	GatewayManage GatewayViewAllDetails GatewayDeploy GatewayRequestDeploy GatewayApproveDeployRequest GatewayGrantManageGateway GatewayGrantViewGateway GatewayGrantDeployAPI GatewayGrantRequestDeployAPI

Grant Name	Description	Can be Issued To	Associated Actions
View all details	People issued this grant are allowed to view all information about this gateway	Gateway Managers, API Managers, Plan Managers	GatewayViewAllDetails
Deploy to Gateway	People issued this grant are allowed to deploy or undeploy APIs to this gateway.	Gateway Managers, API Managers	GatewayDeploy GatewayRequestDeploy
Request Deployment to Gateway	People issued this grant are allowed to request API deployments to this gateway. Requests must be approved by a Gateway Manager	API Managers	GatewayRequestDeploy
Node Service Account	Gateway Runtime service accounts are issued this grant to allow them to download configuration and upload statistics.	GatewayRuntime	GatewayRetrieveConfiguration GatewayUploadStatistics

Application Grants

Grant Name	Description	Can be Issued To	Associated Actions
Manage Application	People issued this grant can view, modify and delete this application. API Manager users issued this grant can also issue grants for this application to others.	API Managers, Application Developers, Plan Managers	ApplicationEdit ApplicationDelete ApplicationView ApplicationGrantManageApplication
View All Details	People issued this grant can see all details about this application in the Developer Portal.	API Managers, Application Developers, Plan Managers	ApplicationViewAllDetails

Service Account Grants

Grant Name	Description	Can be Issued To	Associated Actions
Manage Service Account	People issued this grant are allowed to view, modify and delete this service account.	Service Managers	ServiceAccountEditAll ServiceAccountViewAllDetails ServiceAccountViewHistory ServiceAccountDelete ServiceAccountReference ServiceAccountGrantManageServiceAccount ServiceAccountGrantViewAllDetails ServiceAccountGrantReferenceServiceAccount
View all details	People issued this grant are allowed to see all details about this service account.	API Managers, Gateway Managers, Service Managers	ServiceAccountViewHistory ServiceAccountViewAllDetails
Reference Service Account	People issued this grant are allowed to reference this service account (add it to policies).	API Managers, Service Managers	ServiceAccountViewAllDetails ServiceAccountReference

Service Grants

Grant Name	Description	Can be Issued To	Associated Actions
Manage Service	People issued this grant are allowed to view, modify and delete this service.	Service Managers	ServiceEditAll ServiceModifyState ServiceViewAllDetails ServiceViewHistory ServiceDelete ServiceReference ServiceGrantManageService ServiceGrantViewAllDetails ServiceGrantReferenceService
View All Details	People issued this grant are allowed to see all details about this service.	API Managers, Gateway Managers, Service Managers	ServiceViewAllDetails ServiceViewHistory

Grant Name	Description	Can be Issued To	Associated Actions
Reference Service	Users issued this grant are allowed to reference this service (add it to policies).	API Managers, Service Managers	ServiceViewAllDetails ServiceReference

Plan Grants

Grant Name	Description	Can be Issued to	Associated Actions
Manage the plan	Users issued this grant are allowed to modify the definition of and issue users grants for this plan.	Plan Managers	PlanEditAll PlanEditPublic PlanDelete PlanModifyPublishState PlanModifyState PlanViewAllDetails PlanViewPublicDetails PlanViewHistory PlanRequestSubscribeApplication PlanSubscribeApplication PlanApproveSubscription PlanEntitleAPI PlanGrantViewAllDetails PlanGrantViewPublicDetails PlanGrantManagePlan PlanGrantRequestSubscribeApplication PlanGrantSubscribeApplication PlanGrantEntitleAPI
View all details	Users issued this grant are allowed to view all details of this plan in the Management Portal.	API Managers, Gateway Managers, Plan Managers	PlanViewAllDetails PlanViewPublicDetails PlanViewHistory
View public details	Users issued this grant are allowed to see the public details of this plan in the Developer Portal.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails
Subscribe	Users issued this grant are allowed to subscribe applications to this plan.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails PlanSubscribeApplication

Grant Name	Description	Can be Issued to	Associated Actions
Request Subscription	Users issued this grant are allowed to request to subscribe applications to this plan.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails PlanRequestSubscribeApplication
Entitle	Users issued this grant are allowed to entitle APIs to this plan.	API Managers, Plan Managers	PlanViewPublicDetails PlanEntitleAPI

2

Administering Oracle API Platform Cloud Service - Classic

Review the following topics to manage users, manage roles, update platform settings, and to perform other Oracle API Platform Cloud Service - Classic administration tasks.

Topics

Typical Workflow for Administering Oracle API Platform Cloud Service - Classic

To start administering Oracle API Platform Cloud Service - Classic, refer to the typical task workflow.

Task	Description	More Information
Create users and groups	Add users and groups in your instance's Fusion Middleware Control console.	Managing Users and Groups
Assign roles	Assign roles to users and groups to manage what actions people can perform.	Managing Roles
Configure OAuth providers	Configure OAuth 2.0 providers for tokens to validate services secured by the OAuth 2.0 policy.	Configure OAuth Providers
Manage SSH access	View and manage SSH keys for instances in your identity domain.	Managing SSH Access for Oracle API Platform Cloud Service - Classic Instances
Manage access rules	Create and manage access rules for your instance.	Managing Access Rules for Oracle API Platform Cloud Service - Classic Instances
Start, stop, and restart instances	Start and stop instances and, when service instances are running, start, stop, and restart individual server or load balancer VMs.	Starting, Stopping, and Restarting Oracle API Platform Cloud Service - Classic Instances
Back up and restore instances	Back up and restore your service instances.	Backing Up and Restoring an Oracle API Platform Cloud Service - Classic Instance
Update platform settings	Adjust the default time zone and manage the Developer Portal.	Updating Platform Settings

Best Practices for Managing your Oracle API Platform Cloud Service - Classic Instance

Although you have full access to the VMs your service instance runs on, follow these best practices to ensure you can migrate your Oracle API Platform Cloud Service - Classic instance to newer versions when they are released.

WARNING:

Failing to follow these guidelines may make it difficult or impossible to migrate your service instance to newer versions of the product.

Layered Resources

Oracle API Platform Cloud Service - Classic is built on the following layers:

- **VM Layer**: includes the virtual machine image and the Oracle Linux operating system running in the VM.
- **Platform Layer:** includes the WebLogic Server application server. This layer includes associations with Database Cloud Service (DBCS) and Oracle Storage Service (OSCS).
- **Oracle API Platform Cloud Service - Classic Layer:** This includes the Oracle API Platform Cloud Service - Classic Management and Development Portals (if the Developer Portal is not deployed on-premises).

Supported Administration Actions

These administration tasks are supported:

- Signing in to the Fusion Middleware Control Console for Your Instance
- Signing in to the WebLogic Administration Console for Your Instance
- Signing in to the Load Balancer Console for Your Instance
- Accessing a VM Through a Secure Shell (SSH)
- Managing Roles
- Updating Platform Settings
- Customizing the Look and Feel of the Developer Portal
- Adding or Modifying Developer Portal Language Resources
- Deploying the Developer Portal On Premise
- Set the Time Display
- Configure Accessibility Preferences for Oracle API Platform Cloud Service - Classic
- Configure OAuth Providers
- Managing Access Rules for Oracle API Platform Cloud Service - Classic Instances
- Starting, Stopping, and Restarting Oracle API Platform Cloud Service - Classic Instances
- Backing Up and Restoring an Oracle API Platform Cloud Service - Classic Instance
- Delete an Oracle API Platform Cloud Service - Classic Instance

Unsupported VM Layer Administration Actions

> **WARNING:**
>
> Don’t perform any of the following actions at the VM layer. These actions are unsupported and may make it difficult or impossible to migrate your service instance to newer versions of the product.

- Patch Oracle Linux with unsupported updates
- Tune Oracle Linux operating system settings
- Install or deinstall Oracle Linux programs
- Delete files, including temporary files
- Modify Oracle Linux operating system privileges
- Modify SSH keys

Unsupported Platform Layer Administration Actions

> **WARNING:**
>
> Don't perform any of the following actions at the Platform layer. These actions are unsupported and may make it difficult or impossible to migrate your service instance to newer versions of the product.

- Install custom certificates in the Management Tier

  You can install custom certificates for accessing secured backend services on each gateway node. See Configure SSL Certificates to Pass Requests to Services Over HTTPS.
- Open additional ports on the Management Tier VMs
- Install custom applications to WebLogic Server or to service VMs
- Change the WebLogic Server topology, including adding or deleting managed servers
- Change the WebLogic Server JVM settings
- Create or modify WebLogic Server resources, like data sources, connection pools, JMS servers, work managers, and partitions.
- Modify WebLogic Server MBean settings
- Modify OPSS permissions

Oracle API Platform Cloud Service - Classic Layer Administration Guidelines

> **WARNING:**
>
> Follow these guidelines when creating and managing users and groups in Oracle API Platform Cloud Service - Classic. Failing to follow these guidelines may make it difficult or impossible to migrate your service instance to newer versions of the product.

- **Do use email addresses for user names**: Ensure you use email addresses for all user accounts you create in Fusion Middleware Control for accessing the Oracle API Platform Cloud Service - Classic Management and Developer Portals. See Adding Users .

- **Don’t create nested groups**: Ensure that every group you create in Fusion Middleware Control is not a member of any other group. This includes the pre-seeded role groups. For example, if you create a new Operators group, you should not add this group to an existing group, like APICSAdministrators. See Adding Users to a Group.

Oracle Database Cloud Service Best Practices

Follow these best practices to ensure disk space does not fill up on your Database Cloud Service VMs.

- Clean up the archived logs periodically.
- If you have configured a longer retention policy, increase the storage volume used by the fast recovery area by extending the backup storage volume. See Scaling a Database Deployment in *Using Oracle Database Cloud Service*.

Signing in to the Fusion Middleware Control Console for Your Instance

Use Fusion Middleware Control for your Oracle API Platform Cloud Service - Classic instance to create users and groups and to assign users to groups. You can sign in to Fusion Middleware Control to perform other administrative tasks.

To sign in to the Fusion Middleware Control console:

1. Sign in to the My Services Console. See Accessing the My Services Console.
2. On the Services page, click the  Manage this service icon for your instance, and then click **Open Fusion Middleware Control Console**.
3. When the login page appears, enter the WebLogic Administrator user name and the password you specified when you provisioned your Oracle API Platform Cloud Service - Classic instance, or enter a user name that is granted one of the default global security roles.

Signing in to the WebLogic Administration Console for Your Instance

You can sign in to the WebLogic Administration Console for your Oracle API Platform Cloud Service - Classic instance to perform administrative tasks.

To sign into the WebLogic Administration Console:

1. Sign in to the My Services Console. See Accessing the My Services Console.
2. On the Services page, click the  Manage this service icon for your instance, and then click **Open WebLogic Server Console**.
3. When the login page appears, enter the WebLogic Administrator user name and the password you specified when you provisioned your Oracle API Platform Cloud Service - Classic instance, or enter a user name that is granted one of the default global security roles.

Signing in to the Load Balancer Console for Your Instance

You can sign in to the Oracle Traffic Director Load Balancer Console for your Oracle API Platform Cloud Service - Classic instance to perform load balancer configuration tasks.

To sign into the Load Balancer Console:

1. Sign in to the My Services Console. See Accessing the My Services Console.
2. On the Services page, click the  Manage this service icon for your instance, and then click **Open Load Balancer Console**.
3. When the login page appears, enter the WebLogic Administrator user name and the password you specified when you provisioned your Oracle API Platform Cloud Service - Classic instance.

Accessing a VM Through a Secure Shell (SSH)

You can access the service instance's VMs by logging into the VM as the `opc` user through SSH. You can use any SSH utility you want. For example, if you are using Windows, you might use PuTTY; if you are using Linux, you might use OpenSSH.

> **Note:**
>
> Only the `opc` user can remotely connect to your VMs. You can not use SSH to connect to a VM as the `oracle` user. After successfully connecting to a VM, tasks such as starting and stopping the server and accessing the administrative logs should only be performed by the `oracle` user.

Topics

- Understanding SSH Keys
- Generating a Secure Shell (SSH) Public/Private Key Pair
- Connecting to an Administration Server or Load Balancer VM
- Connecting to a Managed Server VM
- Creating an SSH Tunnel
- Switching VM Users

Understanding SSH Keys

To access an Oracle API Platform Cloud Service - Classic virtual machine (VM) with a secure shell (SSH) client, you must create a public/private key pair and configure the service instance with the public key.

When you create an Oracle API Platform Cloud Service - Classic instance, you are prompted to supply the public key. You can either provide an existing public key that you previously created with an external tool, or Oracle API Platform Cloud Service -

Classic can create a new key pair for you. To connect to a VM in an Oracle API Platform Cloud Service - Classic instance, you supply the paired private key when logging in to the machine using an SSH client.

You may also use the same SSH public/private key pair that you used for creating an Oracle Database Cloud Service database deployment.

See also:

- Creating an Oracle API Platform Cloud Service - Classic Instance
- Generating a Secure Shell (SSH) Public/Private Key Pair
- The Creating SSH Keys for Use with Oracle Cloud Services tutorial
- Managing SSH Access for Oracle API Platform Cloud Service - Classic Instances

Generating a Secure Shell (SSH) Public/Private Key Pair

Several tools exist to generate SSH public/private key pairs. The following sections show how to generate an SSH key pair on UNIX, UNIX-like and Windows platforms.

Topics

- Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility
- Generating an SSH Key Pair on Windows Using the PuTTYgen Program

Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh-keygen utility to generate SSH key pairs.

To generate an SSH key pair on UNIX and UNIX-like platforms using the ssh-keygen utility:

1. Navigate to your home directory:

   ```
   cd $HOME
   ```

2. Run the ssh-keygen utility, providing as *`filename`* your choice of file name for the private key:

   ```
   $ ssh-keygen -b 2048 -t rsa -f filename
   ```

   The ssh-keygen utility prompts you for a passphrase for the private key.

3. Enter a passphrase for the private key, or press Enter to create a private key without a passphrase:

   ```
   Enter passphrase (empty for no passphrase): passphrase
   ```

> **Note:**
>
> While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

The ssh-keygen utility prompts you to enter the passphrase again.

4. Enter the passphrase again, or press Enter again to continue creating a private key without a passphrase:

   ```
   Enter the same passphrase again: passphrase
   ```

5. The ssh-keygen utility displays a message indicating that the private key has been saved as *filename* and the public key has been saved as *filename*`.pub`. It also displays information about the key fingerprint and randomart image.

Generating an SSH Key Pair on Windows Using the PuTTYgen Program

The PuTTYgen program is part of PuTTY, an open source networking client for the Windows platform.

To generate an SSH key pair on Windows using the PuTTYgen program:

1. Download and install PuTTY or PuTTYgen.

   To download PuTTY or PuTTYgen, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTYgen program.

   The PuTTY Key Generator window is displayed.

3. Set the **Type of key to generate** option to **SSH-2 RSA**.

4. In the **Number of bits in a generated key** box, enter `2048`.

5. Click **Generate** to generate a public/private key pair.

   As the key is being generated, move the mouse around the blank area as directed.

6. (Optional) Enter a passphrase for the private key in the **Key passphrase** box and reenter it in the **Confirm passphrase** box.

> **Note:**
>
> While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

7. Click **Save private key** to save the private key to a file. To adhere to file-naming conventions, you should give the private key file an extension of `.ppk` (PuTTY private key).

> **Note:**
>
> The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY as your SSH client. It cannot be used with other SSH client tools. Refer to the PuTTY documentation to convert a private key in this format to a different format.

8. Select all of the characters in the **Public key for pasting into OpenSSH authorized_keys file** box.

   Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.

9. Right click somewhere in the selected text and select **Copy** from the menu.
10. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.
11. Save the text file in the same folder where you saved the private key, using the `.pub` extension to indicate that the file contains a public key.
12. If you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux), export the private key:
    a. On the **Conversions** menu, choose **Export OpenSSH key** .
    b. Save the private key in OpenSSH format in the same folder where you saved the private key in `.ppk` format, using an extension such as `.openssh` to indicate the file's content.

Connecting to an Administration Server or Load Balancer VM

You can access an Administration Server or a Load Balancer VM through a secure shell (SSH) utility.

To access a VM through SSH:

1. Navigate to the Services page of the My Services Console.
2. Click the service instance associated with the VM you want to access.

   The Oracle API Platform Cloud Service - Classic Overview page appears, displaying detailed information about the service instance.

3. From the list of virtual machines, note the **Public IP** address of the Administration Server, the Managed Servers, or the Load Balancer, depending on which VM you want to access.

   This address will be specified in the typical octet format (111.111.111.111).

4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to connect to the VM as the `opc` user.

   Provide the following:

   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The VM's public IP address.

in this format:

```
ssh -i path_to_private_key opc@VM_IP_address
```

For example:

```
ssh -i /home/myuser/id_rsa opc@111.111.111.111
```

5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to connect to the VM as the `opc` user.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

   a. Launch PuTTY.

      The PuTTY Configuration window is displayed, showing the Session panel.

   b. In the **Host Name (or IP address)** field, enter the public IP address of the VM.

   c. In the Category tree, expand **Connection** if necessary and then click **Data**.

   d. In the **Auto-login username** field, enter `opc`.

   e. Confirm that the **When username is not specified** option is set to **Prompt**.

   f. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

   g. Under **Private key file for authentication**, click **Browse**.

   h. Navigate to and select your private key file. Then click **Open**.

      > **Note:**
      >
      > The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY. If you have to use a key saved in a different format, see the PuTTY documentation.

   i. Click **Open** to open the connection to the VM.

6. If the private key was defined with a passphrase, enter this value when prompted.

When the VM command line appears, you can use any resource accessible from the VM. For example, you can run the WebLogic Scripting Tool on the Administration Server VM.

Connecting to a Managed Server VM

You can access a Managed Server VM through a secure shell (SSH) utility by using the Administration Server VM as a proxy.

Alternatively, you can connect to the Administration Server VM with SSH, and from within this SSH session start another SSH connection to the Managed Server VM.

To connect to a Managed Server VM by using the proxy method:

1. Navigate to the My Services Console.
2. Click the service instance associated with the VM you want to access.

The Oracle API Platform Cloud Service - Classic Overview page appears, displaying detailed information about the service instance.

3. From the list of virtual machines, identify the following information:
   - The **Public IP** address of the Administration Server VM (used as the proxy).
   - The **Host** name of the Managed Server VM to which you want to connect.
4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to connect to the VM as the `opc` user.

   Provide the following:
   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The Administration Server VM's public IP address.
   - The Managed Server VM's host name.

   in this format:

   ```
   ssh -i path_to_private_key -o ProxyCommand="ssh -W %h:%p -i
   path_to_private_key opc@admin_server_VM_IP_address"
   opc@managed_server_host_name
   ```

   For example:

   ```
   ssh -i /home/myuser/id_rsa -o ProxyCommand="ssh -W %h:%p -i /home/
   myuser/id_rsa opc@111.111.111.111" opc@myjcs-wls-2
   ```

5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to connect to the VM as the `opc` user.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

   a. Launch PuTTY. If your private key was defined with a passphrase, then you must use the `pageant` utility to launch PuTTY:

      ```
      pageant "path to private key" -c "path to putty"
      ```

      For example:

      ```
      c:\PuTTY\pageant "c:\oracle\rsa.ppk" -c "c:\PuTTY\putty"
      ```

   b. If you used `pageant` to start PuTTY, enter the passphrase for the private key.

      The PuTTY Configuration window is displayed, showing the Session panel.

   c. In the **Host Name (or IP address)** field, enter the host name of the Managed Server VM.

   d. In the Category tree, expand **Connection** if necessary and then click **Data**.

   e. In the **Auto-login username** field, enter `opc`.

   f. Confirm that the **When username is not specified** option is set to **Prompt**.

   g. In the Category tree, click **Connection > Proxy**.

h. Set **Proxy type** to **Local**.

i. In the **Proxy hostname** field, enter the IP address of the Administration Server VM.

j. Set the **Port** to 22.

k. In the **Telnet command or local proxy command** field, enter the following value:

```
plink -i "path to private key" opc@%proxyhost -nc %host:%port
```

For example:

```
plink -i "c:\\oracle\\rsa.ppk" opc@%proxyhost -nc %host:%port
```

l. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

m. Under **Private key file for authentication**, click **Browse**.

n. Navigate to and select your private key file. Then click **Open**.

> **Note:**
>
> The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY. If you have to use a key saved in a different format, see the PuTTY documentation.

o. Click **Open** to open the connection to the VM.

> **Note:**
>
> You can optionally save this session configuration by navigating to the Session panel and clicking **Save**. When you open PuTTY the next time, you can load this configuration by selecting it and clicking **Load**.

When the VM command line appears, you can use any resource accessible from the VM.

Creating an SSH Tunnel

An SSH tunnel to an Oracle API Platform Cloud Service - Classic VM enables you to connect to other non-public ports on the VM though a port your local machine.

If a resource provided by a VM uses a port that is not directly accessible through the Internet, you can access that resource by creating an SSH tunnel to the port.

In general an SSH tunnel may map a remote port to any available port number on your local machine. However, port 9001 on the Administration Server uses JMX/RMI for communication, which requires that the remote and local port numbers be the same value. Therefore, the following instructions configure the tunnel's local port number to the same value as the VM's port number.

Tutorial

To set up an SSH tunnel to a VM:

1. Navigate to the Services page of the My Services Console.
2. Click the service instance associated with the VM you want to access.

   The Oracle API Platform Cloud Service - Classic Overview page appears, displaying detailed information about the service instance.
3. From the list of virtual machines, note the **Public IP** address of the Administration Server, the Managed Servers, or the Load Balancer, depending on which VM you want to access.

   This address will be specified in the typical octet format (111.111.111.111).
4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to create an SSH tunnel to the VM.

   Provide the following:

   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The VM's public IP address.
   - The port number on the VM to which you want to connect. The SSH tunnel will enable connectivity to this remote port though the same port number on your local machine.

   in this format:

   ```
   ssh -i path_to_private_key -L port:VM_IP_address:port opc@VM_IP_address
   -N
   ```

   For example, to create an SSH tunnel to port 9001 on the Administration Server VM:

   ```
   ssh -i /home/myuser/id_rsa -L 9001:111.111.111.111:9001
   opc@111.111.111.111 -N
   ```

5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to create an SSH tunnel to the VM.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

   a. Launch PuTTY.

      The PuTTY Configuration window is displayed, showing the Session panel.

   b. In the **Host Name (or IP address)** field, enter the public IP address of the VM.

   c. In the Category tree, expand **Connection** if necessary and then click **Data**.

   d. In the **Auto-login username** field, enter `opc`.

   e. Confirm that the **When username is not specified** option is set to **Prompt**.

   f. In the Category tree, click **Connection > SSH**.

   g. Under **Protocol options**, select the checkbox **Don't start a shell command at all**.

h. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

i. Under **Private key file for authentication**, click **Browse**.

j. Navigate to and select your private key file. Then click **Open**.

> **Note:**
>
> The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY. If you have to use a key saved in a different format, see the PuTTY documentation.

k. In the Category tree, click **Connection > SSH > Tunnels**.

l. In the **Destination** field, enter *`IP`*`:`*`port`*,

where *IP* is the IP address of the VM and *port* is the port number on the VM to which you want to connect.

m. In the **Source Port** field, enter the same port number.

n. Click the **Add** button.

o. Click **Open** to create the SSH tunnel to the VM.

> **Note:**
>
> You can optionally save this session configuration by navigating to the Session panel and clicking **Save**. When you open PuTTY the next time, you can load this configuration by selecting it and clicking **Load**.

6. If the private key was defined with a passphrase, enter this value when prompted.

Applications running on your local machine can now communicate with the VM by using `localhost:`*`port`*, where *port* is the local port number.

For example, after creating an SSH tunnel to port 9001 on the Administration Server VM, launch a web browser and connect to `http://localhost:9001/console`.

> **Note:**
>
> After your work with the SSH tunnel is complete, perform a `<ctrl> C` to shut down the SSH tunnel.

Switching VM Users

You can change users on an Oracle API Platform Cloud Service - Classic VM in order to perform specific administration tasks.

You must SSH to a VM only as the `opc` user. This user has root privileges on the OS running in the VM. For example, `opc` can be used to create other OS users on a VM. Simply prefix root operations with the `sudo` command. For example:

```
sudo useradd myuser
```

> **Note:**
>
> There is no default password for the `opc` user.

Switching to Oracle

The `oracle` VM user has regular OS user permissions. It is intended to be used to start and stop Oracle products that have been installed on the VM, or to run other Oracle applications and utilities on the VM.

Type the following to become the `oracle` user:

```
sudo su - oracle
```

> **Note:**
>
> There is no default password for the `oracle` user.

Switching to Root

An alternative to using the `sudo` command to perform root OS operations with the `opc` user is to switch to the `root` user.

Type the following to become the `root` user:

```
sudo -s
```

> **Note:**
>
> Avoid using the `root` user except to perform privileged OS administration tasks.

Configuring SSL and Custom Domain Name for Oracle API Platform Cloud Service - Classic

You can configure Secure Socket Layer (SSL) between clients and the nodes in your Oracle API Platform Cloud Service - Classic instance to ensure that applications are accessed securely and configure custom domain name so that your clients can access your applications using a custom domain name instead of a public IP address.

SSL is the most commonly-used method of securing data sent across the internet, and assures visitors that transactions with your application are secure. By default SSL is already enabled within the software components of a service instance. They are configured to use a self-signed SSL certificate that was generated by Oracle API Platform Cloud Service - Classic. Clients will typically receive a message indicating that the signing certificate authority (CA) for this certificate is unknown and not trusted. You can update the load balancers to use a custom SSL certificate, or a certificate that you have obtained from a CA.

By using the load balancer as the front-end, you can quickly and easily associate a custom "vanity" domain name to your application environment. Rather than accessing your applications using a public IP address, clients can use your custom domain name.

Refer to the following topics for information on configuring SSL and custom domain name.

- Configuring SSL for Oracle Java Cloud Service Instance.
- Configuring SSL for the Load Balancer
- Defining a Custom Domain Name for an Oracle Java Cloud Service Instance

Use the SSL Certificate Import Utility

The SSL Certificate Import utility enables you to get the signed certificates of the connecting server and import it into your keystore.

SSL certificates enable secure connections between API boundaries. This is important if you are exposing APIs for consumption or if you are invoking existing APIs. Importing SSL certificates can be a tedious and often error-prone practice. TheOracle API Platform Cloud Service - Classic provides a utility that makes this process easier.

Topics:

About the Utility

Before you run the utility, there are some things you should know.

The utility retrieves all of the necessary certificates from the server automatically and in the proper format. When you provide the name of the keystore to which you want to

import the certificates, be sure to provide the keystore file that belongs to the JDK being used by the Oracle API Platform Cloud Service - Classic gateway. Otherwise, the utility may import the certificates successfully, but there is no effect in runtime during API calls.

About Certificate Types

When you use the SSL Certificate Import Utility, you must specify the certificate type you want to import. There are three options:

- CA — This means CA Certificate chain. Use this option when you want to import the intermediate and root certificates.
- IM — This means Intermediate. Use this option when you want to import the intermediate certificate only.
- SS — This means Self-signed. Use this option when you want to import the intermediate, root, and server certificates.

> **Note:**
>
> The SS option is common in scenarios in which you are accessing backend services that use load balancers with self-signed certificates. Oracle Traffic Director (OTD) is a load balancer that comes with a self-signed certificate by default. This certificate needs to be imported into the Oracle API Platform Cloud Service - Classic gateway to allow the traffic. This includes OTD instances provisioned from the Oracle Cloud, including OTD instances from Java Cloud Service or SOA Cloud Service.

About Certificate Aliases

The utility has an option to specify aliases for each certificate imported. After you specify the certificate type, the utility knows which certificates you are importing, and it will ask you if you want to provide an alias. Answer “y” to the prompt, and it will ask for the test for the alias.

The alias is a plain string but with no spaces and preferably with no special characters. Providing an alias while importing certificates is not a mandatory step but rather a best practice. If you don’t provide one, an machine-generated alias will be created for you. By definition; an alias is just a handle to a key/pair or a certificate, so if you ever need to modify one that has been imported before, you can reference it using a more user-friendly name..

About Network Proxies

The utility must have an outbound internet connection to fetch the certificate(s) from the URL you provide. If you are running the utility from an environment that usually does not provide an outbound internet connection, you won’t be able to run the utility correctly. A common way to overcome this limitation is using network proxies when available.

If you need to use a proxy, you must provide the proxy host and port. The value of the host must be a valid address that points to the network proxy. The utility does not infer if the network proxy is SSL-based or not, so you must provide the appropriate protocol prefix (HTTP or HTTPS) before the address. The port must be a valid, non-negative

number. Usually, it is 80 for HTTP-based network proxies or 443 if it is HTTPS. Make sure to provide the correct values for the network proxy host and port, otherwise the utility will fail.

Download the SSLCertUtility

You must download the SSLCertUtility before you can run it.

The utility is found within the contents of the gateway installer for the Oracle API Platform Cloud Service - Classic.

To download the utility:

1. Log in to the Oracle API Platform Cloud Service - Classic Management Portal.
2. Click the **Gateways** tab.
3. On the Gateways List page, click a gateway.
4. Click the **Nodes** tab for the gateway.
5. Click **Download Gateway Installer**.
6. Follow the steps in your browser to save the zip file to a location.
7. Once the file has finished downloading, extract it to a folder.

Run the SSLCertUtility

After you have downloaded the utility, you can run it to import the certificates.

As you run the utility, it prompts you with a series of questions.

To run the utility:

1. In the directory where you extracted the contents of the zip file you downloaded, open a terminal window.
2. Enter `./sslcertutiltool.sh` and press Enter.
3. Enter the URL of the server from which you want to import the certificates, for example, `https://www.example.com` and press Enter.
4. Enter the location of the keystore to which you want to import the certificates. This should be a full, absolute path to the keystore, such as `example/oracle/jdkl.8.0_161/jre/lib/security/certs`. Press Enter.
5. Enter the keystore password and press Enter.
6. Specify the certificate types you want to import. The options are IM, CA, and SS. Press Enter.
7. Depending on which certificate type you specified, the utility asks if you have an alias for the certificate. If you answer “y” (yes), it then prompts you for the alias.
8. The utility then asks if you need a proxy to connect to the server. If you answer “y” (yes), you are prompted for the proxy host, such as `www-proxy.us.example.com`, and the proxy port.
9. Specify whether you want the utility to run in debug mode. While you can specify either “y” (yes) or “n” (no), yes is recommended, so you can see all the details step-by-step.

The utility creates a backup of your keystore file before doing any changes. The backup is created in the same directory that holds the key store file that you have provided and has the `.backup` extension. If anything goes wrong, you can revert to the backup.

Configuring a Hostname for Oracle API Platform Cloud Service - Classic

Configuring a hostname allows you to use a domain name instead of an IP address to access an Oracle API Platform Cloud Service - Classic instance.

how to get the certificate, how to apply the certificate, and then make the OTD use it as the named server

- Register your domain name through a third-party domain registration vendor, such as `verisign.com`, `register.com` and `namecheap.com`.

  > **Note:**
  >
  > Get a CA (certificate authority)-issued certificate, not an SSL (self-signed) certificate.

- Resolve your domain name to the IP address of the Oracle API Platform Cloud Service - Classic load balancer, using the third-party domain registration vendor console.
- Import the certificate to the load balancer. Once the certificate authority has validated the certificate, import the certificate through the Load Balancer. See Importing a CA-Issued SSL Certificate to the Load Balancer in *Administering Oracle SOA Cloud Service in a Customer-Managed Environment*.
- Associate the certificate to the HTTPS listeners in the load balancer's configuration. The load balancer then presents the SSL certificate while processing any new HTTPS requests. See Associating the SSL Certificate With the Load Balancer
- Update the managed servers to use the domain name rather than redirect to the OTD IP address. Log in to the Admin server where Oracle API Platform Cloud Service - Classic is installed and follow these steps:
  - Under **Environment**, click **Clusters** and click the cluster name.
  - On the **Configuration** tab, click the **HTTP** tab and select **Frontend Host**.
  - Enter the domain name you used when you created the certificate and click **Save**.
  - Activate the changes and restart the managed servers.

> **Note:**
>
> Once the new hostname is configured, the gateway installer (`gateway-props.json` file) should always use the hostname in the `managementServiceURL` property.

Managing Users and Groups

Add users and groups to Oracle API Platform Cloud Service - Classic in your instance's Fusion Middleware Control console. See Signing in to the Fusion Middleware Control Console for Your Instance.

Topics

Adding Users

Add users in the Enterprise Manager Fusion Middleware Control console for your Oracle API Platform Cloud Service - Classic instance.

You must use the WebLogic Administrator account, created when you provisioned your Oracle API Platform Cloud Service - Classic instance, to add users in the Fusion Middleware Control console.

This task assumes that you've already signed in to the Fusion Middleware Control console for your Oracle API Platform Cloud Service - Classic instance. See Signing in to the Fusion Middleware Control Console for Your Instance.

To add Oracle API Platform Cloud Service - Classic users:

1. Expand the **WebLogic Server Domain** menu, select **Security**, and then click **Users and Groups**.
2. On the Users and Groups page, ensure the**Users** tab is displayed.
3. Click **Create**.
4. In the **Name** field, enter a name for the user.

> **Note:**
>
> User names are case insensitive and must be unique. User names must not include any of the following characters:
>
> - Semicolons — ;
> - Commas — ,
> - Plus signs — +
> - Equal signs — =
> - Single backslash character — \ (note that two consecutive backslashes may be used; for example `smith\\`)
>
> User names must not begin with either of the following characters:
>
> - Pound sign — #
> - Double quotation — "

5. **(Optional)** In the **Description** field, enter a short description for the user.
6. In the **Provider** list, ensure that the **DefaultAuthenticator** option is selected.
7. In the **Password** field, enter a password for the user. Confirm by entering the same password in the **Confirm Password** field.
8. Click **Create**.

The user is created.

Adding Groups

Add groups in the Enterprise Manager Fusion Middleware Control console for your Oracle API Platform Cloud Service - Classic instance.

You must use the WebLogic Administrator account, created when you provisioned your Oracle API Platform Cloud Service - Classic instance, to add groups in the Fusion Middleware Control console for your instance.

This task assumes that you've already signed in to the Fusion Middleware Control console for your Oracle API Platform Cloud Service - Classic instance. See Signing in to the Fusion Middleware Control Console for Your Instance.

To add Oracle API Platform Cloud Service - Classic groups:

1. Expand the **WebLogic Server Domain** menu, select **Security**, and then click **Users and Groups**.
2. On the Users and Groups page, click the **Groups** tab.
3. Click **Create**.
4. In the **Name** field, enter a name for the group.

> **Note:**
>
> Group names are case insensitive and must be unique. Group names must not include any of the following characters:
>
> - Angle brackets — < or >
> - Backslashes — \
> - Commas — ,
> - Equals signs — =
> - Forward slashes — /
> - Parentheses — (or)
> - Plus signs — +
> - Question marks — ?
> - Semicolons — ;
> - Square brackets — [or]
>
> Group names must not begin with either of the following characters:
>
> - Pound sign — #
> - Double quotation — "

5. **(Optional)** In the **Description** field, enter a short description for the group.
6. In the **Provider** list, ensure that the **DefaultAuthenticator** option is selected.
7. Click **Create**.

Adding Users to a Group

Add users to a group to give them the same privileges (roles and resource grants) the group possesses.

> **WARNING:**
>
> As a best practice for migrating to newer versions, you should not add groups to other groups. See Best Practices for Managing your Oracle API Platform Cloud Service - Classic Instance.

You also assign roles to users by adding them to the pre-seeded group associated with the role. For example, if you want to assign a user the API Manager role, you must add them to the APIManagers group that is created when you provision your service instance to give them API Management privileges. See Roles.

You must use the WebLogic Administrator account, created when you provisioned your Oracle API Platform Cloud Service - Classic instance, to add groups in the Enterprise Manager Fusion Middleware Control console for your instance.

This task assumes that you've already signed in to the Fusion Middleware Control console for your Oracle API Platform Cloud Service - Classic instance. See Signing in to the Fusion Middleware Control Console for Your Instance.

To add users to a group:

1. Expand the **WebLogic Server Domain** menu, select **Security**, and then click **Users and Groups**.
2. On the Users and Groups page, ensure the **Users** tab is displayed.
3. Click the user you want to add to a group.
4. Click the **Groups** tab.
5. In the **Available** list, select the group(s) you want to add this user to, and then click the **>** (Move selected items to: Chosen) icon to move the groups to the **Chosen** list.

   These groups are created when you provision your service instance and correspond to the specified role:

Group	Associated Role
APICSAdministrators	Administrator
APIGatewayManagers	Gateway Manager
APIGatewayRuntimeUsers	Gateway Runtime
APIManagers	API Manager
APPDevelopers	Application Developer
ServiceManagers	Service Manager

   See Roles.
6. Click **Save**.

Managing Roles

You assign roles to users and groups to manage what actions people can perform in Oracle API Platform Cloud Service - Classic.

In this release, each role corresponds to a group that is created when you create your service instance. To assign a role, you manually map each user to the group associated with each role. See Adding Users to a Group and Roles.

> **Note:**
>
> In addition to roles, you issue grants to specific resources (like an API or a gateway) to refine who can interact with each resource. See Grants.

Viewing Users and Groups Assigned a Role

You can view which users and groups are assigned each role. You must be assigned the Administrator role to complete this task.

1. From the Roles List page, click the role for which you want to view users and groups.

The users and groups assigned the role appear.

Viewing Which Grants Can be Issued to Users or Groups Assigned to a Role

You can view which grants can be issued to users assigned each role. You can also see a description of these grants and the actions they allow users to perform.

1. From the Roles List page, select the role you want to view eligible grants for.
2. Click the  (Grants) tab.

   All of the grants that can be issued to users or groups assigned this role are displayed and described.

> **Tip:**
>
> Click a grant to display the actions users receiving this grant can perform.

Updating Platform Settings

You can adjust the default time zone and manage the Developer Portal from the Platform Settings page in the Management Portal.

Topics

Setting the Default Time Zone

You can set the default time in which API Platform displays API usage and analytics data.

This allows everyone to view the same data regardless of the time zone in which they access the platform. As an example, Susan access the Management Portal from the US west coast and Tony accesses from the US east coast. If the default time zone is set to PST, Susan and Tony view the same API usage data when viewing analytics charts filtered to show data from **Today**, though Tony's "today" starts at 3AM local time due to the time differential.

You must be assigned the Administrator role to set the default time zone.

To set the default time zone:

1. Click the **Platform Settings** tab.

2. On the General Settings page, select the time zone you want to use to display API usage and analytics data from the **Time Zone Settings** list, .
3. Click **Save**.

The default time zone is set. Analytics charts and other data are displayed in this time zone.

Enabling or Disabling the Developer Portal

Use the Platform Settings page to enable or disable the Developer Portal. Users attempting to access the Developer Portal when it is disabled receive a 404--Not Found error. You must be assigned the Administrator role to disable the Developer Portal.

To disable the Developer Portal:

1. Click the **Platform Settings** tab.
2. Click the  (Developer Portal Settings) tab.
3. Click the **Developer Portal** switch to move it to the OFF position.
4. Click **Save**.

The developer portal is now disabled.

> **Note:**
>
> You can enable the Developer Portal by reversing these steps: click the **Developer Portal** switch to move it to the ON position, and then click **Save**.

Changing the Developer Portal URL

Change the Developer Portal URL if you have deployed it on premise. When you change the Developer Portal URL in the Management Portal, publication and preview links are updated to point to this location. You must be assigned the Administrator role to change the Developer Portal URL.

To change the Developer Portal URL

1. Click the **Platform Settings** tab.
2. Click the  (Developer Portal Settings) tab.
3. In the **Portal Server URL Configuration** section, click the **Custom** button.
4. Enter the URL at which the APIs page on the Developer Portal can be accessed, such as `http://example.com:7201/developers/apis`.

> **Note:**
>
> The APIs page is available at `<the Developer Portal base URL/apis>`.

5. Click **Save**.

The Developer Portal URL has been changed. You can see the updated URL on the APIs List page for published APIs and other places throughout the Management Portal.

Hiding or Showing OAuth Client ID and Secret and Application Keys in the Developer Portal

Use the **Client Security** options on the Platform Settings page in the Management Portal to show or hide OAuth client IDs and secrets or application keys in the Developer Portal.

You must be assigned the Administrator role to configure the Client Security options.

1. Click the **Platform Settings** tab.
2. Click the  (Developer Portal Settings) tab.
3. Configure **Client Security** options:
   a. Deselect **Show OAuth 2.0 Client ID and Secret** to hide, or select to display, these properties in the Developer Portal. This option is selected by default.
   b. Deselect **Show Application Key** to hide, or select to display, these properties in the Developer Portal. This option is selected by default.
4. Click **Save**.

The Client Security options you deselected are hidden, and the options you selected are displayed, in the Developer Portal.

Customizing the Look and Feel of the Developer Portal

The Developer Portal branding, general layout, page structure, appearance and behaviors are controlled by a JSON configuration file. You can customize the Developer Portal by editing this file and submitting it via the REST API.
You must be assigned the Administrator role to customize the look and feel of the Developer Portal.

See Sample Developer Portal Configuration File for a sample configuration file. The recommended method to customize the portal is to download the JSON file and change only the objects that you want to modify. The result will be a subset of the original configuration file that you can post to the REST service. The Developer Portal merges your definitions on top of the original configuration file at run-time and modifies the configuration. You can also post a partial JSON file as long as it retains the same structure as the original configuration file.

See *Set configuration* in the *REST API for the Consumer Service in Oracle API Platform Cloud Service* for details about what each line in the configuration file does.

Only users assigned the Administrator role can post a configuration file using the REST API.

1. Download the current JSON configuration file from `<Host where Developer Portal is deployed:port>/developers/oap/apiportal.config.json`.

2. Open the JSON file in a plain text editor and modify the properties you want to change. For example, you can modify the logo, background, product name, and page title. Save the file.
3. Use cURL or the REST client of your choice to post the JSON file:

```
curl -X PUT
--cacert ~/cacert.pem
-u username:password
-d @apiportal.config.json
https://<developer portal location>/developers/services/v1/portal/
customization/configuration
```

   See *Set configuration* in the *REST API for the Consumer Service in Oracle API Platform Cloud Service*.

See *Delete configuration* in the *REST API for the Consumer Service in Oracle API Platform Cloud Service*to revert the Developer Portal to its default appearance.

Sample Developer Portal Configuration File

The sample is the default Developer Portal configuration file included in a fresh Oracle API Platform Cloud Service - Classic deployment.

See the *Set Configuration* operation in the *REST API for the Consumer Service in Oracle API Platform Cloud Service*for details about what each line in the configuration file does.

```
{
  "language": "en",
  "resources": "oap/core/i18n/resources/",
  "branding": {
    "vendor": "${i18n.branding.vendor}",
    "product": "${i18n.branding.product}",
    "product_short": "${i18n.branding.product_short}",
    "title": "${i18n.branding.title}",
    "logo": {
      "url": "oap/css/images/login/OracleLogoBLK.png",
      "width": "auto",
      "height": "17px",
      "alignment": "baseline"
    },
    "login": {
      "logo": {
        "url": "oap/css/images/login/OracleLogoBLK.png",
        "width": "137px"
      },
      "splash": "img/OracleLogoWHT.png",
      "productLogo": "oap/css/images/login/productLogo.png",
      "background": {
        "desktop": "oap/css/images/login/Desktop.jpg",
        "tablet": {
          "portrait": "oap/css/images/login/TabletPT.jpg",
          "landscape": "oap/css/images/login/TabletLS.jpg"
        },
```

```
      "mobile": {
        "portrait": "oap/css/images/login/MobilePT.jpg",
        "landscape": "oap/css/images/login/MobileLS.jpg"
      }
    },
    "css": {
    }
  },
  "icon": {
    "url": "oap/css/images/general/favicon.ico",
    "type": "image/x-icon"
  },
  "about": {
    "body": [
      "${i18n.oap.navbar.about.paragraph1}",
      "${i18n.oap.navbar.about.paragraph2}"
    ],
    "links": [
      {
        "text": "${i18n.oap.navbar.about.links.aboutOracle}",
        "url": "http://www.oracle.com/us/corporate/index.html#menu-about"
      },
      {
        "text": "${i18n.oap.navbar.about.links.contactUs}",
        "url": "http://www.oracle.com/us/corporate/contact/index.html"
      },
      {
        "text": "${i18n.oap.navbar.about.links.legalNotices}",
        "url": "http://www.oracle.com/us/legal/index.html"
      },
      {
        "text": "${i18n.oap.navbar.about.links.termsOfUse}",
        "url": "http://www.oracle.com/us/legal/terms/index.html"
      },
      {
        "text": "${i18n.oap.navbar.about.links.privacyRights}",
        "url": "http://www.oracle.com/us/legal/privacy/index.html"
      }
    ]
  },
  "copyright": "${i18n.branding.copyright}"
},
"css": {
  "*": {
    "font-family": "\"Helvetica Neue\",Helvetica,Arial,sans-serif",
    "font-size": "14px"
  }
},
"services": {
  "portal": "services/portal/v1/",
  "manager": "services/management/v1/",
  "analytics": "services/analytics/v1/",
  "administration": "services/administration/v1/",
  "console": "console/"
},
```

```
  "session": {
    "logout": "${documentBaseUri}/logout",
    "timeout": "${documentBaseUri}/logout?session"
  },
  "modules": {
    "base": "oap/modules/",
    "inventory": {
      "header": {
        "path": "header/",
        "css": {
          "*": {
            "font-size": "1.5em",
            "background-color": "#f0f0f0"
          }
        }
      },
      "navbar": {
        "path": "navbar/",
        "documentationUrl": "http://www.oracle.com/pls/topic/lookup?
ctx=cloud&id=api-platform-cloud-dev-tasks"
      },
      "messaging": {
        "path": "messaging/"
      },
      "error": {
        "path": "error/"
      },
      "login": {
        "path": "login/",
        "urlScheme": "login",
        "loadIndicator": false,
        "fullScreen": "full-height,no-viewport-scaling"
      },
      "api.catalog": {
        "path": "api/catalog/",
        "urlScheme": "apis",
        "pageId": {
          "root": "EB69C5B6-3B51-48E1-91CF-B4E8996A1973"
        }
      },
      "api.details": {
        "path": "api/details/",
        "urlScheme": [
          "apis/{vanityName}",
          "apis/{vanityName}/{iteration:published|current}",
          "apis/{vanityName}/{section:overview|documentation}",
          "apis/{vanityName}/{section:overview|documentation}/
{iteration:published|current}",
          "apis/{vanityName}/{iteration:published|current}/
{section:overview|documentation}"
        ],
        "title": "${api.name} - ${config.branding.title}",
        "pageId": {
          "root": "AA9E8B29-E37D-48A3-A497-45FD5150C722",
          "deep": {
```

```
    "documentation": "BECD3CE5-3945-4A24-AF14-D703A43BC62F"
  }
},
"domain": "api.catalog",
"data": {
  "apiaryTheme": {
    "tableOfContents": {
      "section": {
        "color": "#0572ce",
        "fontFamily": "arial, helvetica, sans serif",
        "fontWeight": "normal",
        "paddingBottom": "10px",
        "title": {
          "padding": "2px 0px",
          "$hover": {
            "backgroundColor": "#e4f0fa",
            "text": {
            }
          }
        },
        "item": {
          "paddingLeft": "16px",
          "borderLeft": "3px solid transparent",
          "text": {
            "margin": "2px 0px 1px 0px"
          },
          "$hover": {
            "backgroundColor": "#e4f0fa",
            "text": {
            }
          },
          "$selected": {
            "borderLeft": "3px solid #0572ce",
            "text": {
            }
          },
          "subitems": {
            "subitem": {
              "borderLeft": "2px solid transparent",
              "$hover": {
                "text": {
                }
              },
              "$selected": {
                "borderLeft": "2px solid #0572ce",
                "text": {
                }
              }
            }
          }
        }
      }
    },
    "humanColumn": {
      "content": {
```

```
"fontFamily": "arial, helvetica, sans serif",
"apiName": {
  "color": "black",
  "fontSize": "36px",
  "fontWeight": "normal"
},
"section": {
  "marginBottom": "20px",
  "title": {
    "text": {
      "color": "#ed813e"
    }
  },
  "apiDescription": {
    "p": {
      "color": "black",
      "fontSize": "15px",
      "lineHeight": "1.1em"
    }
  },
  "resourceGroups": {
    "resourceGroup": {
      "marginTop": "0px",
      "name": {
        "color": "black",
        "fontSize": "24px",
        "fontWeight": "normal",
        "marginBottom": "10px"
      },
      "resources": {
        "resource": {
          "name": {
            "color": "black",
            "fontSize": "28px",
            "fontWeight": "normal"
          },
          "description": {
            "p": {
              "color": "black",
              "fontSize": "15px",
              "lineHeight": "1.1em"
            }
          },
          "actions": {
            "action": {
              "paddingLeft": "0px",
              "description": {
                "p": {
                  "color": "black",
                  "fontSize": "15px",
                  "lineHeight": "1.1em"
                }
              },
              "invitation": {
                "$hover": {
```

```
                    "backgroundColor": "#e4f0fa"
                  },
                  "$selected": {
                    "backgroundColor": "#0572ce"
                  },
                  "$selected$hover": {
                    "backgroundColor": "#0572ce"
                  },
                  "tag": {
                    "$get": {
                      "backgroundColor": "#267db3"
                    },
                    "$post": {
                      "backgroundColor": "#68c182"
                    },
                    "$put": {
                      "backgroundColor": "#fad55c"
                    },
                    "$delete": {
                      "backgroundColor": "#ed6647"
                    },
                    "$patch": {
                      "backgroundColor": "#8561c8"
                    },
                    "$head": {
                      "backgroundColor": "#6ddbdb"
                    },
                    "$options": {
                      "backgroundColor": "#ffb54d"
                    }
                  }
                }
              }
            }
          }
        },
        "description": {
          "color": "black",
          "fontSize": "15px",
          "lineHeight": "1.1em"
        }
      }
    }
  }
},
"machineColumn": {
  "header": {
  },
  "content": {
    "destination": {
      "container": {
        "uriTemplate": {
          "container": {
            "variable": {
```

```
            "color": "#0572ce"
          }
        }
      },
      "method": {
        "color": "white",
        "border-radius": "0px",
        "border": "1px solid #ccc",
        "$get": {
          "color": "white",
          "backgroundColor": "#267db3"
        },
        "$post": {
          "color": "white",
          "backgroundColor": "#68c182"
        },
        "$put": {
          "color": "black",
          "backgroundColor": "#fad55c"
        },
        "$delete": {
          "color": "white",
          "backgroundColor": "#ed6647"
        },
        "$patch": {
          "color": "white",
          "backgroundColor": "#8561c8"
        },
        "$head": {
          "color": "black",
          "backgroundColor": "#6ddbdb"
        },
        "$options": {
          "color": "black",
          "backgroundColor": "#ffb54d"
        }
      }
    },
    "parameters": {
      "list": {
        "parameter": {
          "key": {
            "color": "#0572ce"
          }
        }
      }
    }
  },
  "console": {
    "breadcrumbs": {
      "font-size": "15px",
      "color": "#999",
      "backgroundColor": "#e4f0fa",
      "borderTop": "0px",
```

```
    "borderBottom": "0px",
    "height": "32px",
    "action": {
      "color": "#666"
    }
  },
  "form": {
    "tabs": {
      "buttonGroup": {
        "borderRadius": "2px",
        "border": "1px solid #c4ced7",
        "height": "28px",
        "item": {
          "backgroundColor": "#e4e8ea",
          "color": "black",
          "$selected": {
            "backgroundColor": "#0572ce",
            "color": "white",
            "fontWeight": "normal"
          }
        }
      }
    },
    "headers": {
      "addHeaderButton": {
        "p": {
          "color": "#0572ce"
        }
      }
    },
    "destination": {
      "container": {
        "uriTemplate": {
          "container": {
            "variable": {
              "color": "#0572ce"
            }
          }
        },
        "method": {
          "border-radius": "0px",
          "border": "1px solid #ccc",
          "$get": {
            "color": "white",
            "backgroundColor": "#267db3"
          },
          "$post": {
            "color": "white",
            "backgroundColor": "#68c182"
          },
          "$put": {
            "color": "black",
            "backgroundColor": "#fad55c"
          },
          "$delete": {
```

```
                    "color": "white",
                    "backgroundColor": "#ed6647"
                  },
                  "$patch": {
                    "color": "white",
                    "backgroundColor": "#8561c8"
                  },
                  "$head": {
                    "color": "black",
                    "backgroundColor": "#6ddbdb"
                  },
                  "$options": {
                    "color": "black",
                    "backgroundColor": "#ffb54d"
                  }
                }
              }
            },
            "parameters": {
              "list": {
                "parameter": {
                  "name": {
                    "color": "#0572ce"
                  }
                }
              }
            },
            "buttons": {
              "submit": {
                "button": {
                  "backgroundColor": "#009c38",
                  "text": {
                    "color": "white"
                  }
                }
              },
              "reset": {
                "button": {
                  "backgroundColor": "#e4e8ea",
                  "text": {
                    "color": "black"
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "api.register": {
    "path": "api/register/",
    "urlScheme": "apis/{vanityName}/register",
    "title": "${api.name} - ${config.branding.title}",
```

```
        "domain": "api.catalog",
        "pageId": {
          "root": "982267B8-72E1-4468-BE46-E24D78301665"
        }
      },
      "api.embeddeddoc": {
        "path": "api/embeddeddoc/"
      },
      "application.catalog": {
        "path": "application/catalog/",
        "urlScheme": "applications",
        "pageId": {
          "root": "DE7D5CC2-7FD2-47E0-90A3-84D86B74203C",
          "deep": {
            "create": "8B5CF77F-8C0A-4B72-A295-E96808B2907A"
          }
        }
      },
      "application.details": {
        "path": "application/details/",
        "urlScheme": [
          "applications/{id:number}",
          "applications/{id:number}/{section:overview|registeredapis|
analytics|users}"
        ],
        "domain": "application.catalog",
        "pageId": {
          "root": "792268FC-FCA1-435F-A199-AA4E545C650E",
          "deep": {
            "apis": "ED226B19-251B-4676-82AC-E7E610325059",
            "grants": "43610297-84FA-425D-9F7E-37886D692C18"
          }
        }
      },
      "application.edit": {
        "path": "application/edit/"
      },
      "application.analytics": {
        "path": "application/analytics/",
        "data": {
          "refreshFrequency": 15,
          "ranges": {
            "default": "last24hours",
            "main": {
              "today": {
                "order": 1,
                "title": "${i18n.oap.application.analytics.label.today}",
                "range": {
                  "from": "day.floor(now)",
                  "to": "day.offset(from,1)"
                },
                "granularity": {
                  "unit": "minute",
                  "length": 30
                }
```

```
                },
                "last24hours": {
                  "order": 2,
                  "title": "$
{i18n.oap.application.analytics.label.last24hours}",
                  "range": {
                    "from": "day.offset(now,-1)",
                    "to": "now"
                  },
                  "granularity": {
                    "unit": "minute",
                    "length": 30
                  }
                }
              },
              "other": {
                "currentHour": {
                  "order": 1,
                  "title": "$
{i18n.oap.application.analytics.timecontrol.currentHour}",
                  "range": {
                    "from": "hour.floor(now)",
                    "to": "hour.offset(from,1)"
                  },
                  "granularity": {
                    "unit": "minute",
                    "length": 1
                  }
                },
                "currentWeek": {
                  "order": 2,
                  "title": "$
{i18n.oap.application.analytics.timecontrol.currentWeek}",
                  "range": {
                    "from": "week.floor(now)",
                    "to": "week.offset(from,1)"
                  },
                  "granularity": {
                    "unit": "hour",
                    "length": 6
                  }
                },
                "month": {
                  "order": 3,
                  "title": "$
{i18n.oap.application.analytics.timecontrol.month}",
                  "items": {
                    "current": {
                      "order": 1,
                      "title": "$
{i18n.oap.application.analytics.timecontrol.current}",
                      "range": {
                        "from": "month.floor(now)",
                        "to": "month.offset(from,1)"
                      },
```

```
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "january": {
    "order": 2,
    "title": "${i18n.oap.commonui.months.january}",
    "range": {
      "from": "year.last(date(now.year,1,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "february": {
    "order": 3,
    "title": "${i18n.oap.commonui.months.february}",
    "range": {
      "from": "year.last(date(now.year,2,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "march": {
    "order": 4,
    "title": "${i18n.oap.commonui.months.march}",
    "range": {
      "from": "year.last(date(now.year,3,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "april": {
    "order": 5,
    "title": "${i18n.oap.commonui.months.april}",
    "range": {
      "from": "year.last(date(now.year,4,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "may": {
    "order": 6,
```

```
    "title": "${i18n.oap.commonui.months.may}",
    "range": {
      "from": "year.last(date(now.year,5,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "june": {
    "order": 7,
    "title": "${i18n.oap.commonui.months.june}",
    "range": {
      "from": "year.last(date(now.year,6,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "july": {
    "order": 8,
    "title": "${i18n.oap.commonui.months.july}",
    "range": {
      "from": "year.last(date(now.year,7,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "august": {
    "order": 9,
    "title": "${i18n.oap.commonui.months.august}",
    "range": {
      "from": "year.last(date(now.year,8,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
      "length": 1
    }
  },
  "september": {
    "order": 10,
    "title": "${i18n.oap.commonui.months.september}",
    "range": {
      "from": "year.last(date(now.year,9,1))",
      "to": "month.offset(from,1)"
    },
    "granularity": {
      "unit": "day",
```

```
                        "length": 1
                      }
                    },
                    "october": {
                      "order": 11,
                      "title": "${i18n.oap.commonui.months.october}",
                      "range": {
                        "from": "year.last(date(now.year,10,1))",
                        "to": "month.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "day",
                        "length": 1
                      }
                    },
                    "november": {
                      "order": 12,
                      "title": "${i18n.oap.commonui.months.november}",
                      "range": {
                        "from": "year.last(date(now.year,11,1))",
                        "to": "month.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "day",
                        "length": 1
                      }
                    },
                    "december": {
                      "order": 13,
                      "title": "${i18n.oap.commonui.months.december}",
                      "range": {
                        "from": "year.last(date(now.year,12,1))",
                        "to": "month.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "day",
                        "length": 1
                      }
                    }
                  }
                },
                "year": {
                  "order": 4,
                  "title": "$
{i18n.oap.application.analytics.timecontrol.year}",
                  "items": {
                    "2016": {
                      "order": 1,
                      "title": "2016",
                      "range": {
                        "from": "date(2016,1,1)",
                        "to": "year.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "week",
```

```
                        "length": 1
                      }
                    },
                    "2015": {
                      "order": 2,
                      "title": "2015",
                      "range": {
                        "from": "date(2015,1,1)",
                        "to": "year.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "week",
                        "length": 1
                      }
                    },
                    "2014": {
                      "order": 3,
                      "title": "2014",
                      "range": {
                        "from": "date(2014,1,1)",
                        "to": "year.offset(from,1)"
                      },
                      "granularity": {
                        "unit": "week",
                        "length": 1
                      }
                    }
                  },
                  "last": {
                    "order": 5,
                    "title": "$
{i18n.oap.application.analytics.timecontrol.last}",
                    "items": {
                      "last15minutes": {
                        "order": 1,
                        "title": "$
{i18n.oap.application.analytics.timecontrol.last15minutes}",
                        "range": {
                          "from": "minute.offset(now,-14)",
                          "to": "now"
                        },
                        "granularity": {
                          "unit": "minute",
                          "length": 1
                        }
                      },
                      "last60minutes": {
                        "order": 2,
                        "title": "$
{i18n.oap.application.analytics.timecontrol.last60minutes}",
                        "range": {
                          "from": "hour.offset(now,-1)",
                          "to": "now"
                        },
```

```
                    "granularity": {
                      "unit": "minute",
                      "length": 1
                    }
                  },
                  "last24hours": {
                    "order": 3,
                    "title": "$
{i18n.oap.application.analytics.timecontrol.last24hours}",
                    "range": {
                      "from": "day.offset(now,-1)",
                      "to": "now"
                    },
                    "granularity": {
                      "unit": "minute",
                      "length": 30
                    }
                  },
                  "last7days": {
                    "order": 4,
                    "title": "$
{i18n.oap.application.analytics.timecontrol.last7days}",
                    "range": {
                      "from": "day.offset(day.floor(now),-6)",
                      "to": "day.ceil(now)"
                    },
                    "granularity": {
                      "unit": "hour",
                      "length": 6
                    }
                  },
                  "last30days": {
                    "order": 5,
                    "title": "$
{i18n.oap.application.analytics.timecontrol.last30days}",
                    "range": {
                      "from": "day.offset(day.floor(now),-29)",
                      "to": "day.ceil(now)"
                    },
                    "granularity": {
                      "unit": "day",
                      "length": 1
                    }
                  },
                  "last365days": {
                    "order": 6,
                    "title": "$
{i18n.oap.application.analytics.timecontrol.last365days}",
                    "range": {
                      "from": "day.offset(day.floor(now),-364)",
                      "to": "day.ceil(now)"
                    },
                    "granularity": {
                      "unit": "week",
                      "length": 1
```

```
                    }
                  }
                }
              }
            }
          }
        },
        "pageId": {
          "deep": {
            "general": "CBD80181-7B89-4B34-AEF9-65833F9736CD",
            "errors": "83BB0064-6774-4F39-8B31-4D3CEC84B1E9"
          }
        }
      }
    },
    "redirect": {
    }
  },
  "layout": {
    "home": {
      "module": "api.catalog"
    },
    "panels": {
      "header": {
        "selector": "header",
        "module": "header"
      },
      "main": {
        "selector": "main",
        "module": "${window.location.pathname}",
        "options": {
          "loadIndicator": true,
          "updatePageTitle": true
        }
      },
      "messages": {
        "selector": "main .oap-messaging",
        "module": "messaging"
      }
    },
    "navigation": {
      "panel": "main",
      "trackHistory": true
    }
  },
  "documentation": {
    "url": "http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=APFDV-
GUID-${pageId}"
  },
  "tracers": {
    "i18n": "Error",
    "services": "Error",
    "applications": "Error",
    "apis": "Error"
```

```
  }
}
```

Adding or Modifying Developer Portal Language Resources

The Developer Portal's language resources and strings are controlled by a JSON resources file. You can customize these resources by editing this file (or creating a file for a new language) and submitting it via the REST API. You can modify resources for the default (English) language or add a full translation of resources in a new language of your choice. The Developer Portal merges your definitions on top of the original resource file at run-time and modifies the language resources. You can also post a partial JSON file as long as it retains the same structure as the original resource file. You must be assigned the Administrator role to add or modify Developer Portal language resources.

See *Set language resource* in the *REST API for the Consumer Service in Oracle API Platform Cloud Service* for details about this REST resource.

1. Download the current JSON configuration file from `<Host where Developer Portal is deployed:port>/developers/oap/core/i18n/resources/root/resources.json`.
2. Open the JSON file in a source code editor and modify the parameters after the `oap` object. For example, you can rename the **Help** button **Documentation**.
3. Save and rename the JSON file with a title that identifies the type of changes implemented by the file. For example, a file named `language_DE.json` identifies a file that adds resources in German.
4. Use this command to post the JSON file:

   ```
   curl -X PUT
   -u username:password
   -d @language_resource.json
   https://<developer portal location>/developers/services/v1/portal/
   customization/language/{languageCode}
   ```

5. Sign in to the Developer Portal and confirm the changes have been applied.

See *Delete language resource* in the *REST API for the Consumer Service in Oracle API Platform Cloud Service* to delete a language resource.

Deploying the Developer Portal On Premise

You can deploy the Developer Portal to an application server running in your infrastructure instead of the instance deployed in Oracle's cloud.

Topics

- Learn About On Premise Deployment of the Developer Portal
- Deploy the Developer Portal to an Oracle WebLogic Server Domain

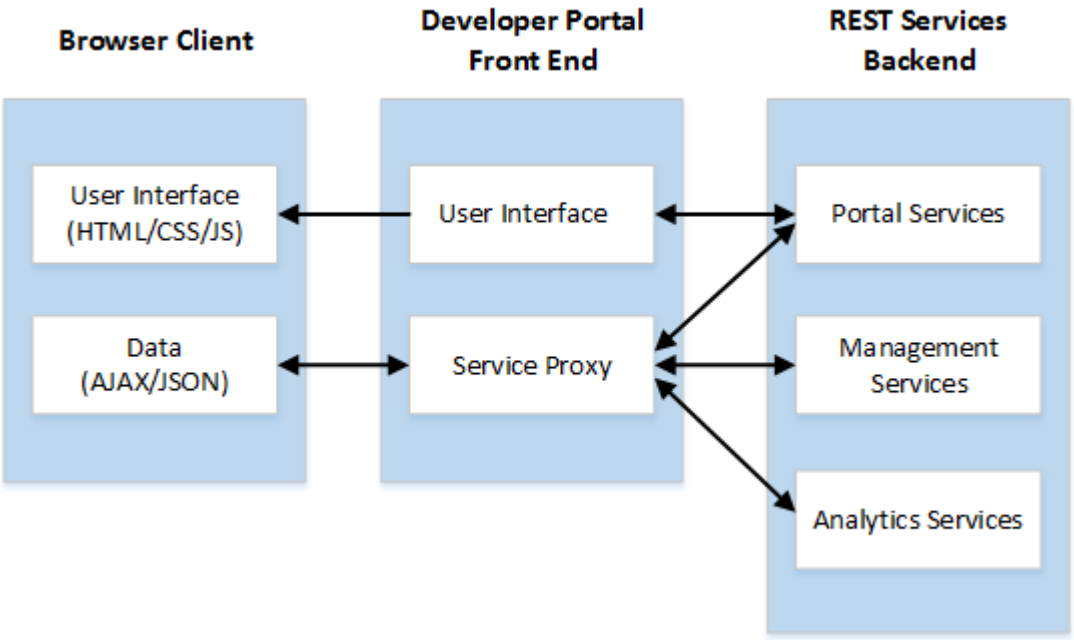
If you do deploy the Developer Portal on-premise, see Changing the Developer Portal URL to update the Management Portal with the correct Development Portal URL for your deployment.

Learn About On Premise Deployment of the Developer Portal

You can deploy the Developer Portal on-premise. The Developer Portal connects to the cloud-based Oracle API Platform Cloud Service - Classic instance through its REST service interface. You specify the base URL of your Oracle API Platform Cloud Service - Classic instance when you prepare the `.war` file for deployment.

Architecture

This image provides an overview of the on premise API Platform Cloud Service Developer Portal architecture:



AJAX data requests submitted on the browser client are sent to the server on which the API Platform Cloud Service Developer Portal user interface is installed. This methodology eliminates the need for Cross-Origin Resource Sharing (CORS) and allows the on premise API Platform Cloud Service Developer Portal to operate without cross-domain issues. AJAX requests are processed by a proxy servlet in the on premise API Platform Cloud Service Developer Portal application.

Access

A login servlet is included with the on premise API Platform Cloud Service Developer Portal application to remove the need for a separate identity system. When you login to the on premise API Platform Cloud Service Developer Portal, their credentials are sent to a backend service for authentication. Basic authentication is used only for the initial, authenticating backend service call. If that succeeds, the cookie sent back by the call is stored for the duration of the session and is used for all subsequent backend calls. Your credentials are not stored.

Deploy the Developer Portal to an Oracle WebLogic Server Domain

You can deploy the Developer Portal to a basic Oracle WebLogic Server 12.2.1 domain. This is currently the only supported application server for Developer Portal deployment.

This task assumes you have already installed Oracle WebLogic Server and created a basic domain. See Creating and Configuring the WebLogic Domain in *Installing and Configuring Oracle WebLogic Server and Coherence*.

You must be assigned the Administrator or Gateway Manager role to download the gateway installer.

To deploy the developer portal to an Oracle WebLogic server domain:

1. Download the Developer Portal WAR (Web application ARchive) file. This file is included in the gateway installer. See Downloading the Gateway Node Installer.
2. Configure the backend service URL:

   You can either configure the web.xml file or the dev-portal.properties file.

   - Configure the web.xml file:

   a. Unzip the installer, navigate to the `developer` directory, and extract the contents of the `oracle.apiplatform.api-portal.war` file.

   b. From the `WEB-INF` directory, edit the `web.xml` file:

   c. Open the `web.xml` file in a plain text editor.

   d. Enter the host and port or IP and port where your Oracle API Platform Cloud Service - Classic instance is accessible within the `<param-value> </param-value>` tags. For example, `<param-value>http://example.com:7201</param-value>` or `<param-value>http://192.168.0.1:7201</param-value>`.

   e. Save and close the file.

   f. Re-archive the contents of the `oracle.apiplatform.api-portal.war` file. Make sure the file extension is still `.war`.

   - Configure the dev-portal.properties file:

   a. From the `apiplatform` directory, edit the `dev-portal.properties` file (this file and directory are user-created).

   b. Open `dev-portal.properties` in a plain text editor.

   c. Add the backendUrl= entry to `dev-portal.properties`. The file is read in initialization time so it must be there when you deploy the application. The domain root is identified by the domain.home Java system property by specifying the `-Ddomain.home=<domain root>` property. If you build your domain manually, the property must be added either in setDomainEnv.sh or by export `EXTRA_JAVA_PROPERTIES="-Ddomain.home=<domain_root>` before you start up the domain.

   d. Save and close the file.

3. Deploy the `oracle.apiplatform.api-portal.war` file to a WebLogic Server domain:

a. Open a command prompt and enter the path to the Oracle WebLogic Server. For example: `C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain\bin`.

b. Run `startWebLogic.cmd` (Windows systems) or `startWebLogic.sh` (Unix-based systems).

c. When the status changes to RUNNING, open a browser and log in to the domain's Administration Console.

d. Click **Deployments** in the **Domain Structure** pane.

e. Click **Install**.

f. Browse to the location of the `oracle.apiplatform.api-portal.war` file containing your edits.

g. Select the `oracle.apiplatform.api-portal.war` file, click **Next**, and complete the Install Application Assistant.

h. Click **Finish**.

The Developer Portal is deployed to the WebLogic Server domain. Try accessing it at `<http or https>://<hostname>:<port>/developers`, where *<hostname>* and *<port>* are the appropriate values for the WebLogic Server domain.

Set the Time Display

By default, times displayed in the Management Portal are displayed using the Platform time zone configured by an administrator. You can choose whether times in the Management Portal are displayed using the Platform time zone or your local time zone.

To set the time display:

1. Click the **User Menu**, and then click **Preferences**.

   The Preferences page appears.

2. Chose one of the time display options:

   - Select **Platform Time Zone** to display all times using the Platform time zone.
   - Select **Local Time Zone** to display all times using your local time zone.

3. Click **Apply**.

The time display option you selected is enabled.

Configure Accessibility Preferences for Oracle API Platform Cloud Service - Classic

You can enable features to make the interface more accessible.

To configure accessibility settings:

1. Click the **User Menu**, and then click **Preferences**.

   The Preferences page appears.

2. Select the accessibility features that you want to enable:

- **High Contrast**: Enables high-contrast in the UI. Select this option, and enable high contrast in your operating system to enable high contrast display.
- **Large Fonts**: Enables large fonts in the UI.

3. Click **Apply**.

The accessibility features you selected are enabled.

Configure OAuth Providers

OAuth 2.0 is an authorization framework that enables an application or service to obtain limited access to a protected HTTP resource. Oracle API Platform Cloud Service - Classic uses OAuth policy to enforce the access token to allow access to protected resources.

Topics

Introduction to OAuth

OAuth is a standard by which a client application can access secure resources without needing username and password credentials. Instead, the client application receives an access token from an OAuth provider which is then used for access to secured resources.

Roles

- Resource Owner: An application or user that can grant access to a resource.
- Resource Server: The API server where the resources are hosted. It can accept and respond to requests that use access tokens.
- Client Application: An application that makes resource requests on behalf of the resource owner.
- Authorization Server: The server that issues access tokens to the client, after the resource owner has been authenticated and authorization is obtained.

Once you have registered an application with the OAuth service, you get a unique client ID and a client secret. The client ID, which is like a username, is for public exposure. It can be included in Javascript source code or be used to create login URLs. The client secret, which is like a password, is used when an application is requesting an access token. The application must know the client secret to receive a token from the authorization server. The client secret must be kept secure.

In addition to validating tokens, access can be limited to APIs using scopes. You can also limit access per HTTP method (GET, PUT, POST, and DELETE) to specific scopes. See Applying OAuth 2.0 Policies

Which OAuth Providers does Oracle API Platform Cloud Service - Classic Support?

This release of Oracle API Platform Cloud Service - Classic can consume tokens from any OAuth provider if the format of the token is JWT, based on RFC7519.

The OAuth Policy asserts the JWT access token and validates various standard claims as defined in RFC7519. The standard claims that are validated are listed below.

- “iss”: Checks that the issuer of the token is valid.
- “sub”: Checks who has created the token.
- “aud”: Checks for whom the token has been created.
- “exp”: Checks the expiration date and time of the token.
- “nbf”: Checks the effective date and time of the token, before which it is not valid.
- “iat”: Checks the issue time of the token.
- “jti”: Checks the unique ID of the token. This is optional.

For more information about the JWT specification, see https://tools.ietf.org/html/rfc7519#section-4.1

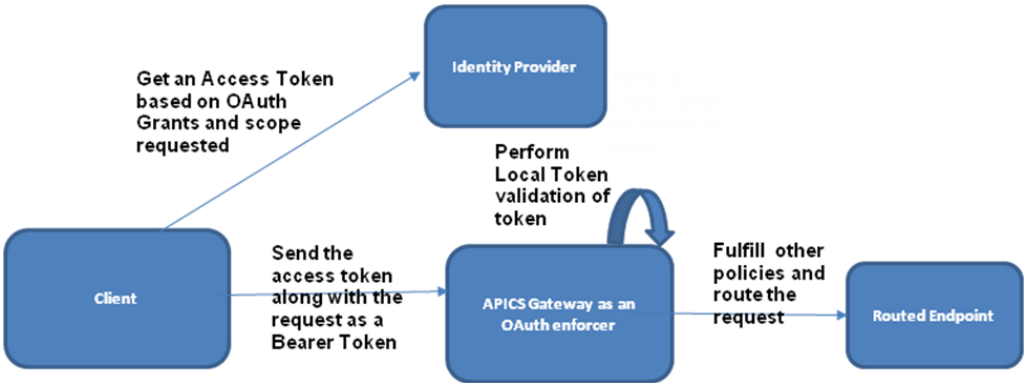
Oracle API Platform Cloud Service - Classic requires that the JWT token be signed per the JWS Compact Serialization format. See https://tools.ietf.org/html/rfc7515#section-3.

Configure the Provider

For any OAuth provider, you create two applications, the resource application and the client application. When you create the resource application, you add the primary audience and the scopes with which the application needs to be protected. When you create the client application, you define the various grant types, such as Resource Owner Credential, Client Credential, Authorization Code, or Implicit, then set the scopes from the Resource Server Application.

OAuth Flow

A client application is authenticated by the identity provider and receives an access token. The client application sends the token to the gateway node, which acts as an OAuth enforcer and validates the token. If the token is valid, the request is passed on to the protected resource.



OAuth Policy Enforcement

The JSON Web Token (JWT) is validated using the following:

- JWT must contain an issuer (“iss”) claim.
- JWT must contain an audience (“aud”) claim.
- JWT must contain an issued (“iss”) and expiry (“exp”) time.
- JWT should be digitally signed to ensure the integrity of the message. The expectation is that it should be signed asymetrically.
- The scope should be defined in the JWT as “scope”. The scope claim is a string with scope claim values separated by spaces. If you have a customized name for the scope claim, you can use the `ScopeClaimName` element in the profile XML file to define it. See The OAuth Profile XML File.

> **Note:**
>
> A subject (“sub”) claim is optional.

Prerequisites

Basic prerequisites:

1. Create an OAuth app on the provider.
2. Create an OAuth client on the provider.

Basic Steps

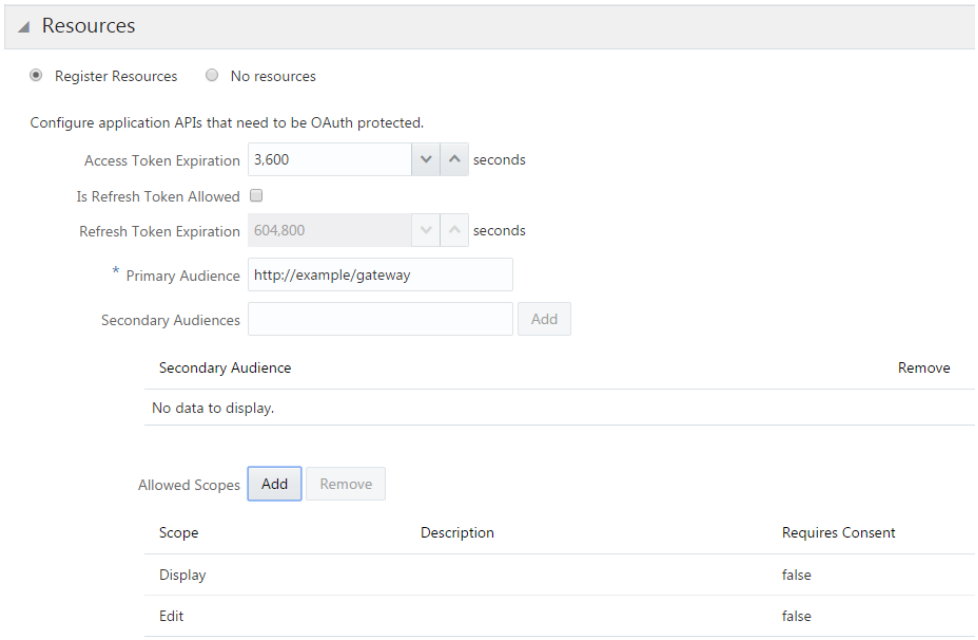
To configure a provider, follow these basic general steps:

1. Create an OAuth resource.
2. Allow the client app to access the resource.
3. Obtain an OAuth token.

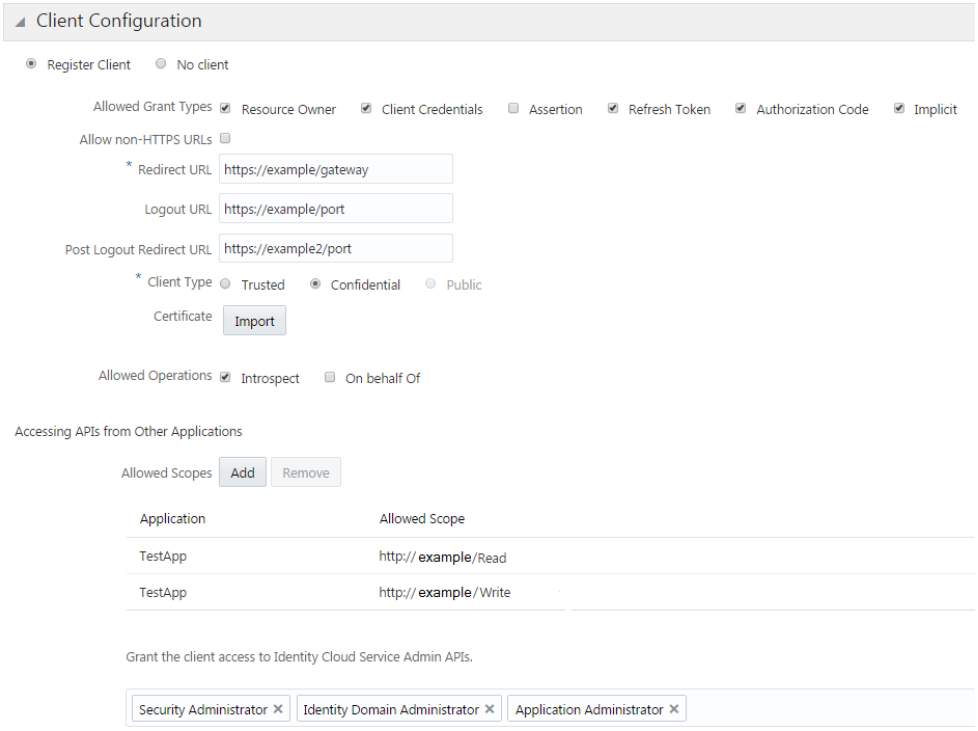
Example Using Oracle API Platform Cloud Service - Classic

If the access token is provided by Oracle Identity Cloud Service, follow these steps to protect a resource.

1. Create a resource server application in Oracle Identity Cloud Service with the primary audience as the endpoint of the resource server. Enter the complete API endpoint, including the load balancing URL.
2. Define scopes for the resource.



3. Create a client application in Oracle Identity Cloud Service with the requisite grants.
4. Add scopes from the resource server application that are allowed for that particular resource. Make sure that the allowed scope that you define matches a scope defined in the application.



Gateway Configuration

1. Create a configuration .xml file. See The OAuth Profile XML File
2. Add the public key of the Oracle Identity Cloud Service instance within the configuration file and configure other mandatory claims for the Oracle Identity Cloud Service instance.

   See Retrieve the Tenant's Signing Certificate in JWK Format in *REST API for Oracle Identity Cloud Service* to get the public key.
3. To protect a resource API, add an OAuth policy at the start of the policy chain. See Applying OAuth 2.0 Policies
4. Deploy the API.

The OAuth Profile XML File

Learn how to configure the OAuth Profile XML file. Gateway nodes use this file to authenticate access tokens clients send with requests to APIs secured by OAuth 2.0 policies.

Upload the OAuth profile to a gateway node via the updateoauthprofile gateway installer action or the *Update Security Profile* operation in the REST API for the Gateway Controller in Oracle API Platform Cloud Service.

Element	Description
`Name`	The type of OAuth profile to use. Supported values are `DEFAULT` and `FORGEROCK_OPENAM`. Oracle Identity Cloud Service can use the DEFAULT profile. The `FORGEROCK_OPENAM` profile is deprecated and will be removed in a future release.
`Issuer`	The identity provider issuer.
`HeaderNameIDToken`	Specify the header clients use to pass ID tokens. This is useful when your OAuth provider creates two tokens: an access token and an ID token. The `HeaderNameIDToken` is deprecated and will be removed in a future release.
`AudienceRestrictionFromConfig`	If set to `true`, audience restriction of the access token is enforced according to the items in the `Audience` element. If set to `false`, audience restriction is enforced through the URI path where the access token is consumed.
`AudienceRestrictionUsingLBR`	If set to `true`, the OAuth policy performs audience restriction validation based on the complete URL where the hostname is the load balancer URL. If set to `false`, the OAuth policy performs audience restriction validation base on the path information where the service is consumed only. It is recommended to set this flag to `true` in a production system. **Note:** The `AudienceRestrictionEntireUrlMatch` element must be set to true if this element is set to true.
`AudienceRestrictionEntireUrlMatch`	If this element is set to true, then the OAuth Policy performs audience restriction validation if the entire consuming URL of the API starts with the audience defined in the claimset.
`Audience`	If `AudienceRestrictionFromConfig` is `true`, the token must contain the audiences listed in this element. Separate multiple audiences with pipes (|).

Element	Description
MandatoryClaims	The list of claims the JWT should contain within the access token.
PublicCertLocation	Includes the public key the gateway uses to authenticate the identity provider. Set the useFormat attribute to specify which type of key is used. PEMFormatPubKey, X509FormatPubKey, JWKFormatPubKey, and NONE are the key types supported in this release.
PEMFormatPubKey	The public key in PEM format. Required only when useFormat is set to PEMFormatPubKey in the PublicCertLocation element.
X509FormatPubKey	The public key in X509 format. Required only when useFormat is set to X509FormatPubKey in the PublicCertLocation element.
JWKFormatPubKey	The public key in JWK format. Required only when useFormat is set to JWKFormatPubKey in the PublicCertLocation element.
	Note: Within the JWKFormatPublicKey there is a kid attribute. This attribute is set to select the appropriate JWK for the JWK set. If this attribute is not defined, then the first JWK is used for the validation of the JWT token.
OutOfBandVerifyAlgorithm	This element is required to restrict JWT to have JWA to RS256 only. If a token is sent with JWA=ES256, the JWT is rejected. Set this element to RS256.
ScopeClaimName	This element is defined to override the default scope claim name from “scope” to a customer defined scope claim name.

Element	Description
`ScopeClaimDataType`	By default, scope values in JWT are space-separated. Scope claim values can also be provided as JSON structure. This element has two valid values: `SPACE_SEPARATED_VALUES` and `JSON`. By default, scope values in JWT are space-separated, so the default value is `SPACE_SEPARATED_VALUES` . If the Scope claim values has json structure, then the value of `ScopeClaimDataType` should be set to JSON..

There are three additional features to note, as described below.

- Allow JWA=NONE over HTTPS channel

  This feature allows JWT to be asserted without validating the signature of the JWT. To prevent Man-in-the-Middle attacks, the JWT must be sent on a secure channel only. If the JWT is not sent over a secure channel, the JWT is rejected. To allow JWT=None, the `useFormat` attribute of the element `PublicCertLocation` should be set to `NONE`.

  ```
  <PublicCertLocation useFormat="NONE">
  </PublicCertLocation>
  ```

- Restrict JWA to RS256 only

  This feature is required to restrict JWT to have JWA to RS256 only. If a token is sent with JWA=ES256, the JWT is rejected. To implement this feature, the OAuth Profile needs to be set the following XML element.

  ```
  <OutOfBandVerifyAlgorithm>RS256</OutOfBandVerifyAlgorithm>
  ```

- Within the OAuth policy, JSON web keys can be selected based on the KID (key ID) that is present in the JSON web token. If the KID is not present in the JSON web token for any reason, it is possible to set the KID as part of the OAuth profile itself. This feature can also be used to override the KID that comes as part of the JSON token.

Sample OAuth Profile

Edit this sample OAuth Profile XML file to match your OAuth implementation before uploading it to gateway nodes.

See the updateoauthprofile gateway installer action or the *Update Security Profile* operation in the REST API for the Gateway Controller in Oracle API Platform Cloud Service.

```
<OAuth2TokenLocalEnforcerConfig>
    <Name>DEFAULT</Name>
    <HeaderNameIDToken>IDToken</HeaderNameIDToken>
    <Issuer>https://identity.oraclecloud.com/</Issuer>
    <AudienceRestrictionFromConfig>true</AudienceRestrictionFromConfig>
    <Audience>http://example:8001|OAuthTestApp</Audience>
    <MandatoryClaims></MandatoryClaims>
    <!-- useFormat has 2 values  PEMFormatPubKey, X509FormatPubKey -->
    <PublicCertLocation useFormat='X509FormatPubKey'>
```

```
<X509FormatPubKey>MIICUDCCAbmgAwIBAgIELfGcXDANBgkqhkiG9w0BAQUFADBXMRMwEQYKCZImiZPyLGQ
BGRYDY29tMRYwFAYKCZImiZPyLGQBGRYGb3JhY2xlMRUwEwYKCZImiZPyLGQBGRYFY2xvdWQxETAPBgNVBAMT
CENsb3VkOUNBMB4XDTE1MTEyMDA5MzI0OFoXDTI1MTExNzA5MzI0OFowXzETMBEGCgmSJomT8ixkARkWA2Nvb
TEWMBQGCgmSJomT8ixkARkWBm9yYWNsZTEVMBMGCgmSJomT8ixkARkWBWNsb3VkMRkwFwYDVQQDDBBvcmNsTV
QxMjMyMzJfaWRtMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCLVvyue+qFraxwM5LxaNLt2QH3wHn/
n0+yk2jmP7mpYkz1xrKuEk2e2SCggzK8MT9jJ5VUaNlF0MwhIZ8/naxA5LPCzGEVfZ/
41GPtGNADFyspqGHkdsNv
+M2eCBme7MDp9L3noBtt2peqGqxSu0DHyt1wgNr6p6EXqTT4AbLdyQIDAQABoyEwHzAdBgNVHQ4EFgQU2rtog
HKC0/ws2dS3Zq7s9wwMofkwDQYJKoZIhvcNAQEFBQADgYEAK1jtcbRpYFAl2Bp9X02MaA/
igq3WXykizH7uQvrWgNQluf7ADbxaB7J96jaIN2GLQFxl6cbPwOvBIu7xd9a26eK6F5gq4iJKm7GeOgV5PZ4r
5umvSZgA0aLOAbhZ/gwy40RauF0X
+4I7JqamnV0DizM2YEDsFWKfTSvCy90ZizMwggJeMIIBx6ADAgECAgRgdcJQMA0GCSqGSIb3DQEBBQUAMFcxE
zARBgoJkiaJk/IsZAEZFgNjb20xFjAUBgoJkiaJk/IsZAEZFgZvcmFjbGUxFTATBgoJkiaJk/
IsZAEZFgVjbG91ZDERMA8GA1UEAxMIQ2xvdWQ5Q0EwIBcNMTUxMTE5MTIwMDQyWhgPMjExNTEwMjYxMTAwNDJ
aMFcxEzARBgoJkiaJk/IsZAEZFgNjb20xFjAUBgoJkiaJk/IsZAEZFgZvcmFjbGUxFTATBgoJkiaJk/
IsZAEZFgVjbG91ZDERMA8GA1UEAxMIQ2xvdWQ5Q0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJeXcnR
eWfVIhqdedQKy+qi+tMp2NstgxHWisJ6OZf7O+KC9WzP+X+UwiQTMUzp
+B4UWUEbpGNW7dv7jiyKdsgYGrnpwINl6OT4wrlKD8r2+7yQLNBsvDQjVeWmhHTqFfgqTfd/wi3MC2itft
+I40O4GnSL3/VDcTSxJZMaigKizAgMBAAGjNTAzMBIGA1UdEwEB/
wQIMAYBAf8CAQAwHQYDVR0OBBYEFPwqb7nQaKEVc
+oSa7ohvOxIGMfXMA0GCSqGSIb3DQEBBQUAA4GBADydG93z/CGRVfUyVfD/ULT/d+RYfm3sriGgPPZlEO
+idCgA6tEMcnIOFf3lP5CFAdFq6ykmFHMOf15CYvqkv7jZULiL3zMgy7OgB4I0i0WMUAAybqeiZlU90y86zd2
yfAgQEM4ncUCHg+Dwf2XF0qmYK0XLF7CSb/hSEJJp4W2j</X509FormatPubKey>
    </PublicCertLocation>
</OAuth2TokenLocalEnforcerConfig>
```

Managing SSH Access for Oracle API Platform Cloud Service - Classic Instances

Use the SSH Access page to view and manage SSH keys for Oracle API Platform Cloud Service - Classic instances in your identity domain.

See also Generating a Secure Shell (SSH) Public/Private Key Pair.

If the SSH private key that you use to access a service instance becomes lost or corrupted, you can add a new public key to the service instance. You may also need to add a new public key to a service instance in order to comply with your organization's security policies or regulations.

1. From the My Services Console click the **SSH Access** tab.
2. Locate your service instance and click **Add New Key**. See Add an SSH Public Key in *Administering PaaS Services*.

See Accessing a VM Through a Secure Shell (SSH) and SSH Access Page.

SSH Access Page

The SSH Access page enables you to view and add SSH public keys to Oracle API Platform Cloud Service instances in your identity domain. You can restrict the list of instances displayed using search filters.

Topics

- What You Can Do From the SSH Access Page
- What You See on the SSH Access Page

What You Can Do From the SSH Access Page

Use the SSH Access page to view and add SSH public keys to Oracle API Platform Cloud Service instances in your identity domain.

You can use the page's Search section to filter the list of displayed deployments based on deployment name.

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the **Expand** icon (the triangle) at the start of a deployment's row to see more details.

What You See on the SSH Access Page

The following table describes the key information shown on the SSH Access page.

Element	Description
Service Name	Filters the results to include SSH keys only for the specified deployment. You can enter a full or partial instance name.
Service Type	Filters the results to include SSH keys only for deployments of the specified service type.
Search	Searches for SSH keys by applying the filters specified by the Service Name and Service Type fields, and displays the results in the table.
Results per page	Specifies the number of results you want to view per page. The default value is `10`.
	Displays a description of an item in the results table. Clicking on the resulting downward arrow hides the description.
Service Name	Shows the name of the deployment.
Service Type	Shows the type of cloud service for this deployment.
Last Update	Shows the most recent time the SSH keys for this deployment were updated.
Actions	Click the **Add New Key** button to add a new SSH public key to this deployment. The **Add New Key** overlay is displayed with its **Key value** field displaying the deployment's most recent SSH public key. Specify the new public key using one of the following methods: • Select **Upload a new SSH Public Key value** and click **Choose File** to select a file that contains the public key. • Select **Key value**. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.

Managing Access Rules for Oracle API Platform Cloud Service - Classic Instances

You can create and manage access rules from the My Services Console and the Oracle API Platform Cloud Service - Classic Overview page.

Access rules enable you to control access to the VMs that make up your service instance. For example, you can enable the database to use a an available port to

access the VM for the WebLogic Administration Server for your service instance. The system creates default rules such as access on port 22 from the public internet to the WebLogic Administration Server VM.

To create access rules for your Oracle API Platform Cloud Service - Classic instance:

1. Navigate to the My Services Console.
2. Click the  **Menu** icon adjacent to the service instance name and select **Access Rules**.

   The Access Rules page is displayed, showing the list of all access rules.
3. Click **Create Rule**.
4. Specify a unique name for the access rule.

   The name must begin with a letter, and can contain numbers, hyphens, or underscores. The length cannot exceed 50 characters. When you create a rule, you cannot use prefixes `ora_` or `sys_`.
5. **(Optional)** Specify a description of the rule
6. Specify a source for the rule:
   - `PUBLIC-INTERNET` —Any host on the internet
   - `WLS_ADMIN_SERVER` — The WebLogic Server Administration Server
   - `WLS_MANAGED_SERVER` — A WebLogic Server Managed Server
   - `PAAS_INFRA` — Internal for platform services. Used for various life cycle operations.
   - `Load Balancer` — The load balancer for the service. The load balancer is identified in the list by its internal name. Example: `_100004-1505196282206`.
   - *`<identifier>`*`:DB` — The database specified when the Oracle API Platform Cloud Service - Classic instance was created.
   - `custom` — A custom list of addresses from which traffic should be allowed. In the field that displays below when you select this option, enter a comma-separated list of the subnets (in CIDR format, such as 192.123.42.1/24) or IPv4 addresses for which you want to permit access.

   The source and the destination must be different.
7. Choose a destination for the rule:
   - `WLS_ADMIN_SERVER` — The WebLogic Server Administration Server
   - `WLS_MANAGED_SERVER` — A WebLogic Server Managed Server
   - `Load Balancer` — The load balancer for the service. The load balancer is identified in the list by its internal name. Example: `_100004-1505196282206`.

   The source and the destination must be different.
8. Specify a port or ports through which the source will access the destination. You can specify a single port or a range of ports (such as 7001–8001).
9. Specify the transport protocol (TCP or UDP) with which the source will access the destination.

10. Click **Create**.

11. To manage your access rules on the Access Rules page, click the **Menu** icon and choose an option:

    - **Enable** — Rules of type USER or DEFAULT can be enabled. Rules of type SYSTEM cannot
    - **Disable** — Rules of type USER or DEFAULT can be disabled. Rules of type SYSTEM cannot.
    - **Delete** — Rules of type USER can be deleted. Rules of type DEFAULT or SYSTEM cannot.

Default Oracle API Platform Cloud Service - Classic Access Rules

The following table describes the default access rules that are created when you create an Oracle API Platform Cloud Service - Classic Instance.

Rule Name	Dafault Status	Ports	Protocol	Source	Destination	Description	Rule Type	Application
sys_ms2db_dblistener	Enabled	1521	tcp	WLS_MANAGED_SERVER	DBaaS:<dbaas_instance>:DB	DO NOT MODIFY: Permit listener connection to database from managed servers	SYSTEM	-
sys_ms2db_ssh	Enabled	22	tcp	WLS_MANAGED_SERVER	DBaaS:<dbaas_instance>:DB	DO NOT MODIFY: Permit managed servers to ssh to db	SYSTEM	-
ora_lb2wls_8001_1	Enabled	8001	tcp	<Load Balancer IPs>	WLS_MANAGED_SERVER	Do not edit or remove: Permit http connection to wls from load balancer	DEFAULT	-
ora_lb2admin_server_7001_1	Enabled	7001	tcp	<Load Balancer IPs>	WLS_ADMIN_SERVER	Do not edit or remove: Permit http connection to admin_server from load balancer	DEFAULT	-

Rule Name	Dafault Status	Ports	Protocol	Source	Destination	Description	Rule Type	Application
ora_p2admin_ssh	Enabled	22	tcp	PUBLIC-INTERNET	WLS_ADMIN_SERVER	Permit public to ssh to admin server	DEFAULT	-
ora_p2admin_ahttps	Disabled	7002	tcp	PUBLIC-INTERNET	WLS_ADMIN_SERVER	Permit public to https to admin server	DEFAULT	-
sys_infra2admin_ssh	Enabled	22	tcp	PUBLIC INTERNET	WLS_ADMIN_SERVER	DO NOT MODIFY: Permit PSM to ssh to admin server	SYSTEM	-

Default Oracle API Platform Cloud Service - Classic Security Lists

The following table describes the default security lists that are created when you create an Oracle API Platform Cloud Service - Classic Instance.

Security List Name	Account
wls/ora_admin	/opcapics/default
wls/ora_ms	/opcapics/default
wls/ora_wls_infraadmin	/opcapics/default
lb/ora_otd	/opcapics/default
lb/ora_otd_infraadmin	/opcapics/default

Default Oracle API Platform Cloud Service - Classic Security Applications

The following table describes the default security applications that are created when you create an Oracle API Platform Cloud Service - Classic instance.

Name	Protocol	Port	Description
sys_chttp	TCP	9073	**DO NOT MODIFY**: Permit HTTP connection to managed servers from OTD
sys_chttps	TCP	9074	**DO NOT MODIFY**: Permits HTTP connection to managed servers from OTD

Name	Protocol	Port	Description
sys_dblistener	TCP	1521	**DO NOT MODIFY**. Permits listener connection to database from managed servers.
wls/ora_ahttps	TCP	7002	Permits traffic from the public internet over HTTPS to the Administration Server.
lb/ora_ahttps	TCP	8989	Permits public to https to OTD admin server
lb/ora_chttps	TCP	443	Permits traffic from the public internet over HTTPS to the Managed Servers.

Starting, Stopping, and Restarting Oracle API Platform Cloud Service - Classic Instances

You can stop and start Oracle API Platform Cloud Service - Classic instances and, when service instances are running, start, stop, and restart individual server or load balancer VMs.

Topics

About Starting, Stopping, and Restarting Oracle API Platform Cloud Service - Classic Instances

You can stop and start an Oracle API Platform Cloud Service - Classic instance and, when the service instance is running, stop, start, and restart individual server or load balancer VMs.

> **Note:**
>
> The stop and restart procedures stop VMs. If you want to shut down the WebLogic Administration Server or Managed Server processes running on the VMs, without stopping the VMs, see Shutting Down and Starting the WebLogic Server Managed Servers and Administration Server Processes on VMs in *Using Oracle Java Cloud Service*. You might want to do this if you have other processes besides the servers running on the VMs and you do not want to shut down these other processes.

This topic describes why you would want to stop or start a service instance, or stop, start, or restart individual server or load balancer VMs. This topic also describes what happens when service instances are stopped and started, and how to monitor these operations.

> **Note:**
>
> You can't start or stop Administration Server VMs; you can restart them instead while the service instance is running. See Restarting the Administration Server VM.

Why Stop an Oracle API Platform Cloud Service - Classic Instance

Stopping an Oracle API Platform Cloud Service - Classic instance frees up compute resources used by the service instance's VMs. Metering for those resources stops.

What Happens When an Oracle API Platform Cloud Service - Classic Instance is Stopped or Started

Stopping and starting an Oracle API Platform Cloud Service - Classic instance has the following results:

- **Stopping the service instance**: The VMs on which the administration server, managed servers, and load balancer are running are stopped.
- **Starting the service instance**: All VMs on which the administration server, managed server and load balancer are running are started. You can restart the administration server, and stop, start, or restart the managed servers and load balancer VMs individually.

> **Note:**
>
> Block storage should not be added manually by using the Oracle Compute Cloud Service because VM restart detaches that block storage. To reattach the block storage, you must use the Oracle Compute Cloud Service. However, block storage added manually is not deleted when an Oracle API Platform Cloud Service - Classic instance is restarted. You must delete it manually.

Why Stop, Start, or Restart an Administration Server, Managed Server, or Load Balancer VM

If an Oracle API Platform Cloud Service - Classic instance is running:

- You can restart the VMs on which the Administration Server, Managed Server, or load balancer are running if you are experiencing problems with the server that would warrant a reboot. The restart operation is the same as stopping the server or load balancer VM, then starting it immediately.
- You can stop the VMs on which the Managed Server or the load balancer are running to free up resources and stop metering those resources. You might also want to stop the service instance instead of scaling, keeping the server or load balancer ready for a later time. If you stop all but one Managed Server VM, you might want to stop the load balancer VM because it is not needed.
- You can start a Managed Server or load balancer VM if it is stopped and you want to use it again. Metering begins again.

How Do I Monitor the Stop, Start, or Restart Operation

You can monitor progress of a stop, start, or restart operation on the Activity section of the Services page.

The Activity section indicates what kind of operation is in progress, and whether it is in progress or complete. When the operation ends, the start and end time of the operation is displayed.

Starting and Stopping an Oracle API Platform Cloud Service - Classic Instance

You can stop and start an Oracle API Platform Cloud Service - Classic instance through the Oracle API Platform Cloud Service - Classic Services page or the Overview page.

To stop or start an Oracle API Platform Cloud Service - Classic instance:

1. Navigate to the Oracle API Platform Cloud Service - Classic Overview page:
   a. From the Services page, click the name of the Oracle API Platform Cloud Service - Classic instance you want to stop or start.

      The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
2. Click the Menu icon in the right corner of the page and select **Start** or **Stop**.

   A confirmation dialog is displayed.
3. Click **OK** in the confirmation dialog.

   When the operation completes, the Oracle API Platform Cloud Service - Classic instance is stopped or started. The entry for the service shows that the stop or start operation has ended.

Restarting the Administration Server VM

You can restart the VM on which the Administration Server is running in an Oracle API Platform Cloud Service - Classic instance that is in a running state.

To restart the Administration Server:

1. Navigate to the Services page, and then click the name of the service instance in which you want to restart the Administration Server.

   The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

2. Click the Menu icon adjacent to the Administration Server row and select **Restart**.

   A confirmation dialog is displayed.

3. Click **OK** in the confirmation dialog.

   The Administration Server VM restarts.

Stoppping, Starting, and Restarting Managed Server and Load Balancer VMs

You can stop, start, and restart the VMs on which the Managed Servers or the load balancer are running in an Oracle API Platform Cloud Service - Classic instance if the service instance is in a running state. Restarting a Managed Server or load balancer VM is the same as stopping it, then starting it.

1. Navigate to the Services page, and then click the name of the service instance in which you want to start, stop, or restart a Managed Server or load balancer VM.

   The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

2. Click the Menu icon to the right of the Managed Server or load balancer row and select **Stop**, **Start**, or **Restart**.

   A confirmation dialog is displayed.

3. Click **OK** in the confirmation dialog.

   The Managed Server or load balancer VM is stopped, started, or restarted.

Restarting WebLogic Servers

You can start and stop servers in an Oracle API Platform Cloud Service - Classic instance by using the WebLogic Server Administration Console and via WebLogic Scripting Tool (WLST) commands.

Oracle API Platform Cloud Service - Classic is built on top of Oracle Java Cloud Service, which in turn is built on top of Oracle WebLogic Server. When you create an API Platform instance, an Oracle WebLogic domain is provisioned across all machines that are part of that instance.

An Oracle WebLogic Domain is made up of a set of WebLogic server instances that work together to host and operate your Java EE applications. Within the domain only one WebLogic server instance is responsible for administrative operations, such as creating new server instances or deploying applications. That privileged server is referred to as the administration server, whereas all the rest are managed servers.

The administration server also hosts the WebLogic Server Administration Console.

The tutorial Restarting WebLogic Servers in an Oracle Java Cloud Service Instance describes how start and stop both the administration and the managed servers in two ways:

- Using the WebLogic Server Administration Console
- Using WLST Commands

Backing Up and Restoring an Oracle API Platform Cloud Service - Classic Instance

Backing up and restoring Oracle API Platform Cloud Service - Classic instances consist of two steps: backing up/restoring the DBaaS instance linked to the API Platform instance and the Oracle API Platform Cloud Service - Classic itself. The back up and restore topics provide the proper order for these steps.

Topics

Backing Up an Oracle API Platform Cloud Service - Classic Instance

You can create a backup immediately without having to wait for the next scheduled backup.

Create a backup when making major changes to your service instance, for example, in these situations:

- Before any configuration changes that you may need to undo
- Before deploying an application
- After deploying an application

> **Note:**
>
> Do **not** to attempt to start the administration server while a backup is in progress.

Note that some configuration details, like access rules, security lists, and security applications, may not be backed up and may need to be restored manually after you restore an instance from a backup.

To initiate an on-demand backup of an Oracle API Platform Cloud Service - Classic instance:

1. Navigate to the Oracle API Platform Cloud Service - Classic Backup page.
   a. From the Services page, click the name of the service instance for which you want to initiate an on-demand backup.

      The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

   b. Click the **Administration** tile.

      The Oracle API Platform Cloud Service - Classic Overview page is refreshed with the Administration tile in focus. The Backup page is displayed.
2. Click **Back Up Now**.

   The Back Up Now dialog box opens.
3. **(Optional)** In the **Back Up Now** dialog box, enter notes about the backup into the **Notes** field.

   You can enter up to 255 characters of free-form text to provide additional information about the backup. This text is displayed in the **Notes** field for the backup in the list of available backups.
4. Click **Back Up**.

   The Backup page is updated to show that the backup is in progress.
   While the backup is in progress, you cannot start any other management operation on the service instance.

   When the backup is complete, it is added to the list of available backups on the Backup page.
5. Back up the DBaaS instance associated with your Oracle API Platform Cloud Service instance. See Creating an On-Demand Backup in *Using Oracle Database Cloud Service*.

Restoring an Oracle API Platform Cloud Service - Classic Instance from Backup

You can restore an Oracle API Platform Cloud Service - Classic instance from a backup to return the service instance to a particular state or recover the service instance after a loss of data.

> **Note:**
>
> You can’t restore an instance from a backup if backups are disabled. Re-enable backups before attempting to restore from a backup.

Note that some configuration details, like access rules, security lists, and security applications, may not be backed up and may need to be restored manually after you restore an instance from a backup.

To restore an Oracle API Platform Cloud Service - Classic instance from a backup:

1. Restore the DBaaS instance associated with your Oracle API Platform Cloud Service - Classic instance. See Restoring from a Specific Backup in *Using Oracle Database Cloud Service*.
2. Navigate to the Oracle API Platform Cloud Service - Classic Backup page.
   a. From the Services page, click the name of the service instance that you want to restore from a backup.

      The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

   b. Click the **Administration** tile.

      The Oracle API Platform Cloud Service - Classic Overview page is refreshed with the Administration tile in focus.

   c. Click the **Backup** tab.

      The Backup page is displayed.
3. From the  menu for the backup in the list of available backups, choose **Restore**.

   A dialog box in which to set options for restoring from the backup opens.
4. In the dialog box, enter notes about the restore operation, and then click **Restore**.

> **Note:**
>
> You **cannot** use Oracle API Platform Cloud Service - Classic to restore the database. To restore the database, you must use Oracle Database Cloud Service to restore from the associated database backup as identified by its RMAN tag. For instructions, see Restoring from a Specific Backup in *Using Oracle Database Cloud Service*.

The Backup page is updated to show that the restoration is in progress. While the restoration is in progress, you cannot start any other management operation on the service instance.

When the restoration is complete, it is added to the restoration history in the Activity log.

Deleting a Backup

You can delete a backup that you no longer require to release storage or prevent an Oracle API Platform Cloud Service - Classic instance from being restored from the backup.

To delete a backup:

1. Navigate to the Oracle API Platform Cloud Service - Classic Backup page.

a. From the Services page, click the name of the service instance for which you want to delete a backup.

   The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

b. Click the **Administration** tile.

   The Oracle API Platform Cloud Service - Classic Overview page is refreshed with the Administration tile in focus.

c. Click the **Backup** tab.

   The Backup page is displayed.

2. From the  menu for the backup in the list of available backups, choose **Delete**.
3. When prompted, confirm that you want to delete the backup.

Disabling Backups

You can disable backups from the My Services Console.

> **Note:**
>
> You can't restore instances from previous backups when backups are disabled. Re-enable backups before attempting to restore from a backup.

To disable backups:

1. Navigate to the Oracle API Platform Cloud Service - Classic Backup page.

   a. From the Services page, click the name of the service instance for which you want to delete a backup.

      The Oracle API Platform Cloud Service - Classic Overview page is displayed with the Overview tile in focus, displaying detailed information about the service instance.

   b. Click the **Administration** tile.

      The Oracle API Platform Cloud Service - Classic Overview page is refreshed with the Administration tile in focus.

   c. Click the **Backup** tab.

      The Backup page is displayed.

2. Click **Disable Backups**.
3. Click **Disable Backups** on the dialog to confirm.

Backups are disabled for this service instance.
To re-enable backups, click **Enable Backups**, and then click **Enable Backups** on the dialog.

Updating and Patching an Instance

You perform an upgrade or patch from the virtual machine (VM) hosting the administration server. You can access this VM through an SSH connection using an account with administrative privileges to run the upgrade and patch scripts. You will need to modify some environmental values in the scripts before running, but these changes are mostly instance specific. As part of provisioning, the scripts for upgrades and patches are deployed to the administration server VM. Each time a subsequent new release is created, or if a patch is needed, the necessary binaries and schema SQL scripts are uploaded into cloud storage for the new version or patch. Using the scripts deployed on the administration server VM, you can download the new binaries and SQL scripts and upgrade or patch the current version.

Topics

Upgrading Gateway Nodes to Release 18.1.5

If you are upgrading from release 18.1.3 to release 18.1.5, the gateway node upgrade is not supported.

To use 18.1.5 features for the gateway nodes, you must install new gateway nodes using the 18.1.5 `ApicsGatewayInstaller.zip` and join to the same logical gateways. The older gateway nodes will continue functioning for the API traffic and the generated analytics will be uploaded to the management service. No new API deployments or subscriptions will be pushed to the gateway nodes.

See Installing a Gateway Node and Register a Node to a Logical Gateway for more information.

You must also turn off polling manually for all nodes in the system before starting the management tier upgrade. Turning off polling does not affect the analytics push.

Updating a Provisioned Instance

For all provisioned instances, the update script location, middleware home, and gateway installer locations are the same.

Do not modify any scripts unless specifically noted.

Provisioned Instance Defaults

The values are stored in `setEnv.sh` of the update scripts.

Value	Location
Update Script	`/u01/app/oracle/tools/paas/state/homes/oracle/run/update`
Middleware Home	`/u01/app/oracle/middleware`
Gateway Installer	`/u01/app/oracle/tools/paas/state/homes/oracle/run/downloads`
API Platform Home	`/u01/app/oracle/suite/apip`

Example 2-1 Default values in setEnv.sh

```
# patch location variables, editable
APICS_SCRIPT_DIR=/u01/app/oracle/tools/paas/state/homes/oracle/run/update
export APICS_SCRIPT_DIR
.
# standard environment variables
GATEWAY_INSTALLER_LOC=/u01/app/oracle/tools/paas/state/homes/oracle/run/
downloads
export GATEWAY_INSTALLER_LOC

MW_HOME=/u01/app/oracle/middleware
export MW_HOME

# APIP_ORACLE_HOME=${MW_HOME}/../suite
export APIP_ORACLE_HOME
```

Instance-Specific Values

Update the setEnv.sh script with instance specific values for the Storage URL, Management Portal URL, and the WLS administrator user name.

These values must be modified for the update to succeed.

User Values

The values are stored in setEnv.sh of the update scripts.

- Storage Location URL — This is the location where the upgrade files are stored in the cloud, `https://storage.us2.oraclecloud.com/v1/jaascdc1-usoracledevop09525/jcs/APICS`.
- Management Portal URL — This is the endpoint for the API Platform Cloud Service Manager Portal, for example `https://example/apiplatform`.
- Weblogic Administrator — Created during original provisioning.
- DBA User (user name for the associated DBaaS service)

Example 2-2 User values in setEnv.sh

```
# storage URL where upgrade bits are stored
STORAGE_URL=<storageUrl>
export STORAGE_URL
```

```
# URL for API Manager Portal
MGR_URL=<managerPortalUrl>
export MGR_URL

# WLS administrator created at initial provisioning
WLS_ADMIN=<wlsAdmin>
export WLS_ADMIN

DBA_USER_NAME=<dbaUser>
export DBA_USER_NAME
```

Pre-Update Steps

Perform these pre-upgrade steps before upgrading your instance.

1. Download the `apics_suite.zip` file for the version you're upgrading to (such as 18.1.3) from the registered PCAR in cloud storage.
2. Make a backup of the original `/u01/app/oracle/tools/paas/state/homes/oracle/run/update` folder.
3. Unzip the contents of the `scripts/update` folder contained in `apics_suite.zip`. It is very important that the `/u01/app/oracle/tools/paas/state/homes/oracle/run/update` folder contains the scripts from `scripts/update` in `apics_suite.zip` or the upgrade process will fail.

   Note: The `update` folder must be moved from the `scripts` folder to the `run` folder.
4. Verify all the shell scripts (*.sh) permissions allow execute permissions.

Update Process

Steps to perform the upgrade or patch:

1. Confirm that a backup of the API Platform Cloud Service and DBaaS instances were performed.
2. Provision an instance script for future upgrade or patch to be deployed to the administration server virtual machine (VM).
3. Connect to the administration server VM through an SSH connection.

   ```
   $ ssh -i privateKey opc@xxx.xxx.x.x
   ```
4. Change to the `oracle` user for administrative privileges:

   ```
   $ sudo su - oracle
   ```
5. Change to the folder where the update scripts reside:

   ```
   cd /u01/app/oracle/tools/paas/state/homes/oracle/run/update
   ```
6. Make updates to `setEnv.sh`.

7. Launch the upgrade or patch script with necessary parameters (upgrade version or patch name and administrative user password).

```
# for upgrades

$ ./upgrade.sh <version>

# example:

$ ./upgrade.sh 18.1.3.0.0_25
```

The scripts execute the following actions:

- Stops and restarts servers.

  Creates a folder with the update script folder with the name of the upgrade version or patch name
- Downloads upgrade or patch package from cloud storage.
- Unzips the upgrade or patch package
- Checks to see if there are any schema update SQL scripts
  - Stops the managed servers if an update SQL script exists
  - Runs the update SQL scripts using the rcuJDBCEngine utility
  - Restarts the managed servers
- Checks to see if there are any seeding SQL scripts for the artifacts and policies
  - Runs the seeding SQL scripts for artifacts and policies
- Checks to see if there are any updated binaries to be deployed (always for upgrades)
- Copies new binaries into middleware home apip/lib folder
- Stops and restarts the administration server and two managed servers

Update is complete. Complete any post-upgrade steps as necessary.

Upgrade logs are located on the administration server where the scripts are run at the directory `/tmp/APICS_<timestamp>`. If you need to view the logs, then when the upgrade completes, look at the last line, for example:

```
Upgrade complete
[oracle@<admin_server_name>upg-wls-1 update]$
```

To see the logs, navigate to `/tmp/APICS_2018-03-16T18:18:25` on the administrative server.

Post Upgrade Steps

Perform these steps to complete a successful upgrade.

Perform these steps as necessary.

- Clean browser cache: To render the landing pages of both the Management and Developer Portals or Oracle API Platform Cloud Service - Classic correctly after upgrade, clean the browser cache and open a new browser window. You may need to do this twice.

- Restart gateway node: To generate analytics from the EDR files, check whether there is TOKEN_ISSUER_URL exception in `/<gateway node domain>/apics/logs/analytics.log`. If it exists, restart the gateway node domain.
- Messages not reflected in Analytics UI: If, after upgrading, messages are not reflected in the Analytics UI, perform the following tasks:
  - Delete the timestamps file to force reprocessing. The command is `rm <gateway-domain-name>/apics/analytics/timestamps`.
  - Check `apics/analytics/logstash.out` for the following string: `PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target`. To resolve the issue, restart `logstash`.

    ```
    cd apics/analytics
    ./manageLogstash.sh stop
    ```

    Wait for 30–60 seconds for logstash process to exit.

    ```
    ./manageLogstash.sh start
    ```

    > **Note:**
    >
    > The messages that are sent after upgrade and before the fix do not appear in the Analytics UI.

Update Folder Structure

The provisioned instance comes with the update scripts deployed in `/u01/app/oracle/tools/paas/state/homes/oracle/run/update` on the administration server VM.

```
$ ls
copyAppProps.py    deployApps.py            getStripeId.sh
rollbackApps.sh         shutdownTarget.sh
copyApps.py        deployApps.sh            patch.sh
rollbackProps.json      startTargetProps.py
copyApps.sh        getSchemaVersion.groovy  readme.txt
setEnv.sh               startTarget.py
createRollback.sh  getSchemaVersion.sh      rollbackAppProps.py
shutdownTargetProps.py  startTarget.sh
deployAppProps.py  getStripeId.groovy       rollbackApps.py
shutdownTarget.py       upgrade.sh
```

Other than the scripts mentioned above, the other scripts should not be modified. They are used in support of the upgrade and patch scripts.

Update Package Structure

The upgrade or patch package used for updating Oracle API Platform Cloud Service - Classic has the following structure:

```
upgrade.zip or <patchName>.zip

    deployProps.json - list of updated web applications to be deployed
    artifacts - folder containing seeding SQL scripts for the artifacts
        sql
            installer
                <seeding scripts for artifacts>
    gatewayInstaller - folder containing updated version of the Gateway
Installer
        ApicsGatewayInstaller.zip - gateway installer
    lib
        <web application binaries>
    policies
        sql
            installer
                <seeding scripts for policies>
    sql
        apip - root folder for APIP schema updates
            apipSeeding.sh - script to launch artifact and policy seeding
SQL scripts for APIP schema
            apipUpgrade.sh - script to launch schema upgrade SQL scripts
for APIP schema
            upgrade - folder containing upgrade SQL scripts for APIP schema
                upgrade.properties - properties file associating upgrade
SQL script to schema version and APIPCS version for APIP schema
                <schema upgrade SQL scripts> - name format is
upgrade_<APIPCS version>.sql
                seeding - folder containing the SQL scripts referencing
the SQL scripts that reside in artifacts and policies root folders
                    scripts
                        insertArtifacts.sql - launch SQL script for
calling artifact seeding SQL scripts
                        insertPolicies.sql - launch SQL script for calling
policy seeding SQL scripts
        apis - root folder for APIS schema updates
            apisSeeding.sh - script to launch artifact and policy seeding
SQL scripts for APIS schema
            apisUpgrade.sh - script to launch schema upgrade SQL scripts
for APIS schema
            upgrade - folder containing upgrade SQL scripts for APIS schema
                upgrade.properties - properties file associating upgrade
SQL script to schema version and APIPCS version for APIS schema
                <schema upgrade SQL scripts> - name format is
upgrade_<APIPCS version>.sql
                seeding - folder containing the SQL scripts referencing
the SQL scripts that reside in artifacts and policies root folders
                    scripts
                        insertArtifacts.sql - launch SQL script for
```

```
calling artifact seeding SQL scripts
                        insertPolicies.sql - launch SQL script for calling
policy seeding SQL scripts
```

Other than the scripts mentioned above, the other scripts should not be modified; They are used in support of the upgrade and patch scripts.

Example Folder Structure

Schemas do not necessarily have to upgrade with each upgrade of APIPCS, so there may not be a corresponding schema upgrade SQL script for the Oracle API Platform Cloud Service - Classic version. Once the upgrade or patch script is run, all work is done within the update script folder. The following shows an example of the folder structure if upgrading to version 17.2.5.0.0_20:

```
# contents of /u01/app/oracle/tools/paas/state/homes/oracle/run/update
17.2.5.0.0_20
copyAppProps.py
copyApps.py
copyApps.sh
createRollback.sh
deployAppProps.py
deployApps.py
deployApps.sh
getSchemaVersion.groovy
getSchemaVersion.sh
getStripeId.groovy
getStripeId.sh
patch.sh
readme.txt
rollbackAppProps.py
rollbackApps.py
rollbackApps.sh
rollbackProps.json
setEnv.sh
shutdownTargetProps.py
shutdownTarget.py
shutdownTarget.sh
startTargetProps.py
startTarget.py
startTarget.sh
test.txt
upgrade.sh

./17.2.5.0.0_20:
artifacts
deployProps.json
gatewayInstaller
lib
policies
sql
upgrade.zip

./17.2.5.0.0_20/artifacts:
sql
```

```
./17.2.5.0.0_20/artifacts/sql:
installer

./17.2.5.0.0_20/artifacts/sql/installer:
apics-analytics.logstash.config.zip.sql
   .
   .
oracle.apiplatform.analytics.agent.ear.tenant.sql

./17.2.5.0.0_20/gatewayInstaller:
ApicsGatewayInstaller.zip

./17.2.5.0.0_20/lib:
oracle.apiplatform.admin.app.ear
   .
   .
oracle.apiplatform.system.app.ear

./17.2.5.0.0_20/policies:
sql

./17.2.5.0.0_20/policies/sql:
installer

./17.2.5.0.0_20/policies/sql/installer:
oracle.apiplatform.policies.apiratelimiting.sql
   .
   .
oracle.apiplatform.policies.serviceresponse.tenant.sql

./17.2.5.0.0_20/sql:
apip
apis

./17.2.5.0.0_20/sql/apip:
apipSeeding.sh
apipUpgrade.sh
upgrade

./17.2.5.0.0_20/sql/apip/upgrade:
seeding
upgrade.properties
upgradeSchema_17.2.1.sql
upgradeSchema_17.2.3.sql

./17.2.5.0.0_20/sql/apip/upgrade/seeding:
scripts

./17.2.5.0.0_20/sql/apip/upgrade/seeding/scripts:
artifacts
policies

./17.2.5.0.0_20/sql/apip/upgrade/seeding/scripts/artifacts:
insertArtifacts.sql
```

```
./17.2.5.0.0_20/sql/apip/upgrade/seeding/scripts/policies:
insertPolicies.sql

./17.2.5.0.0_20/sql/apis:
apisSeeding.sh
apisUpgrade.sh
upgrade

./17.2.5.0.0_20/sql/apis/upgrade:
seeding
upgrade.properties
upgradeSchema_17.2.1.sql

./17.2.5.0.0_20/sql/apis/upgrade/seeding:
scripts

./17.2.5.0.0_20/sql/apis/upgrade/seeding/scripts:
artifacts
policies

./17.2.5.0.0_20/sql/apis/upgrade/seeding/scripts/artifacts:
insertArtifacts.sql

./17.2.5.0.0_20/sql/apis/upgrade/seeding/scripts/policies:
insertPolicies.sql
```

Delete an Oracle API Platform Cloud Service - Classic Instance

When you no longer require an Oracle API Platform Cloud Service - Classic instance, you can delete it.

When you delete an instance, everything is deleted, *including backups*.

To delete an Oracle API Platform Cloud Service - Classic instance:

1. From the menu for the service instance on the Oracle API Platform Cloud Service - Classic Services page, select **Delete**.
2. Enter the database administrator user name and password.
3. **(Optional)** Set the **Force Delete** value to `true` to force the removal of the service instance even if the database instance cannot be reached to delete the database schemas. If set to `true`, you may need to delete the associated database schemas manually on the database instance if they are not deleted as part of the service instance delete operation.

   This value defaults to `true`.
4. **(Optional)** Set the **Should backup be skipped** value to `true` to skip backing up the service instance before deleting it.

   This value defaults to `true`.
5. Click **Delete**.

The database itself is not deleted. Only the repository and schemas created for the Oracle API Platform Cloud Service - Classic instance are deleted.

Once deleted, the Oracle API Platform Cloud Service instance is removed from the list of service instances displayed on the Services page and storage and OCPUs are released.

3
Managing Gateways

Gateway Managers are responsible for managing gateways, the runtime aspect of Oracle API Platform Cloud Service - Classic. Users must be assigned the Administrator or Gateway Manager role and must be issued the required gateway grants to perform actions described in this chapter.

Topics

Typical Workflow for Managing Gateways with Oracle API Platform Cloud Service - Classic

To start managing gateways with Oracle API Platform Cloud Service - Classic, refer to the typical task workflow.

Task	Description	More Information
Read about logical gateways and gateway nodes	Understand the difference between logical gateways and gateway nodes before you design your gateway implementation.	Understanding Gateways and Gateway Nodes
Install gateway nodes	Install a gateway node on-premise, configure and start the domain, create a new logical gateway with the management service, and then register the node to this logical gateway.	Installing a Gateway Node
Configure your gateway node domains	Configure authentication providers and SSL certificates for passing requests to HTTPS endpoints and lock down your nodes.	Configuring Gateway Node Domains
Enable analytics in production environments	Configure each gateway node you install to send analytics data to the management tier.	Enable Analytics in Production Environments
Manage gateway settings	View and manage logical gateway properties, including firewall settings for all nodes registered to the gateway.	Managing Gateway Settings
Manage gateway nodes	Manage registration, polling intervals, and proxies for each of your gateway nodes.	Manage Gateway Nodes in the API Platform Cloud Service Management Portal
Manage gateway grants	Issue fine-grained permissions to users or groups for specific gateways.	Managing Gateway Grants
Work with deployed APIs	View deployed APIs' details, deploy, redeploy, or undeploy APIs, and approve or reject deployment requests.	Working with Deployed Endpoints

Understanding Gateways and Gateway Nodes

A Logical Gateway (called a Gateway in the Management Portal user interface) is a JSON object that defines what its registered nodes should look like. This JSON object resides on the management tier. The definition of each gateway lists the deployed endpoints and the policies applied for each. A gateway node is the physical gateway runtime installation. Gateway nodes can be installed on-premises or in the cloud. Installation of gateway nodes on the same server as the management tier is not supported.

Logical gateways and gateway nodes have a one to many relationship: many nodes can register to one logical gateway, but a node can register to only one logical gateway. Each gateway node polls the management service at configurable intervals to retrieve the JSON logical gateway definition it registers to. See Changing the Node Polling Interval. The node is updated to match the definition. The only period where nodes registered to the same logical gateway are out of sync, except if a given node is

down, is if one has polled the management service and updated based on an updated definition and another has not yet polled.

Because you deploy APIs to logical gateways, and not to gateway nodes, all nodes registered to a gateway have the same APIs deployed with the same policies applied. API Managers can consider using the Gateway-Based Routing Policy (see Applying Gateway-Based Routing Policies) to route to different backend services based on which gateway the API is deployed to; otherwise, if you need your nodes to have different API deployments or different policy configuration, you must create separate logical gateways for each configuration you need.

System Requirements for On-Premises Gateway Installation

The machines you install gateway nodes onto must meet or exceed the requirements listed below.

Component	Requirement
Operating Systems	Oracle Linux and Red Hat Enterprise Linux 7.2, 6.7, and 6.4.
CPU	Dual core, 2 GHz or more per CPU
Disk Space	30 GB
Memory	4 GB
JDK version	Oracle-Certified Java SE JDK 8. OpenJDK is not supported.

Gateway Node Topologies

This release of Oracle API Platform Cloud Service - Classic supports a single gateway node topology.
When you install a gateway node, you create a domain with a single admin server and a single managed server. The gateway node domain uses a lightweight Java database.

This topology is designed to allow you to place multiple lightweight nodes behind a load balancer.

Creating a Logical Gateway

Use the Management Portal or the `createGateway` action in the gateway installer to create a logical gateway.

This task describes how to create a logical gateway. See Understanding Gateways and Gateway Nodes for a description of the relationship between logical gateways and gateway nodes. To install a gateway node, see Installing a Gateway Node.

> **Tip:**
>
> You can also create a logical gateway with the REST API for the Management Service in Oracle API Platform Cloud Service.

Option 1: Create a Logical Gateway in the Management Portal

Perform these steps in the Management Portal UI with a user with the Administrator or the Gateway manager role:

1. From the **Gateways** tab, click **Create Gateway**.
2. In the **Gateway Name** field, enter a name for the logical gateway.
3. **(Optional)** In the **Description** field, describe the logical gateway you're creating.
4. Click **Create**.

   The logical gateway is created. It appears on the Gateway List page. The user who created the gateway is issued the Manage Gateway grant for it.
5. Click the logical gateway you just created. The gateway's Settings page, which displays the gateway's ID and other details, appears.
6. Provide the following details about your gateway, as applicable:
   - **Load Balancer URL**: If gateway nodes registered to this logical gateway are or will be placed behind a load balancer, provide HTTP and HTTPS URLs for the load balancer in the following format: `http://<hostname or IP>:<port>/` for HTTP; `https://<hostname or IP>:<port>/` for HTTPS.
   - **Location**: Describe the location of the nodes registered to this logical gateway.
   - **Firewall**: Configure firewall properties for nodes registered to this logical gateway.

     See Configuring Gateway Firewall Properties.

Option 2: Create a Logical Gateway with the Gateway Node Installer

Perform these steps from a machine you've installed a gateway node to:

1. Edit or add the following properties in the `gateway-props.json` file with details describing the gateway and the management tier:
   - `nodeInstallDir`
   - `logicalGateway`
   - `gatewayNodeName`
   - `managementServiceUrl`

   > **Note:**
   >
   > Note that you may also need to provide values for the `managementServiceConnectionProxy` property and other properties depending on your environment. See gateway-props.json File
2. Run the `creategateway` action.

   When you run this action, you are prompted for the following user credentials:
   - **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

See creategateway.

The logical gateway is created on the Management Portal. The Gateway Manager user you specified is issued the Manage Gateway grant for the gateway and the Gateway Runtime user you specified is issued the Node Service Account for the gateway.

Downloading the Gateway Node Installer

Download the gateway node installer from your Oracle API Platform Cloud Service - Classic Management Portal instance.

To download the gateway installer:

1. Sign into the Management Portal with a Gateway Manager or Administrator user.
2. Click the **Gateways** tab.
3. Click an existing gateway.
4. Click the **Nodes** tab.
5. Click **Download Gateway Installer**. Save the zip file to your local disk.
6. Extract the contents of the zip file into a new directory.

See Installing a Gateway Node to install a node.

Installing a Gateway Node

Install gateway nodes on-premise or in the Cloud, and then register them with a logical gateway on the Oracle API Platform Cloud Service - Classic Management Portal.

> **Note:**
>
> Gateway nodes should not be installed on the same server as the management tier. This type of configuration is not supported.

Topics

- Prerequisites to Install a Gateway Node
- Installing the First Gateway Node for a Logical Gateway
- Installing Additional Gateway Nodes for a Logical Gateway
- Create a New Logical Gateway while Installing a Gateway Node

- gateway-props.json File
- Gateway Node Installer Actions

Prerequisites to Install a Gateway Node

Make sure that the machine on which you are going to install a gateway node meets requirements.

- Ensure your machine meets the minimum requirements. See System Requirements for On-Premises Gateway Installation.
- More than 10GB should be allocated for tmp files.
- The tmp directory should not be set up with `noexec`, `nosuid`, and `nodev`.
- Download the gateway node installer to the machine you want to install a node on. See Downloading the Gateway Node Installer.
- Ensure your *JAVA_HOME* environment variable is set to a supported JDK.
- Set JAVA_HOME to root and exclude the bin folder. For example, if the JAVA_HOME to be used is `/usr/java/` and the java binary is in `/usr/java/bin`, set JAVA_HOME environment variable to `/usr/java`.
- Ensure you have created Gateway Manager and Gateway Runtime users in your identity domain and assigned them appropriate roles. You will need to provide credentials for these users when you install a gateway node. See Adding Users .

> **Note:**
>
> Installing more than one node on a single machine is not supported. Install any additional nodes on separate machines.

> **Note:**
>
> Gateway nodes should not be installed on the same server as the management tier.

Installing the First Gateway Node for a Logical Gateway

You can generate the gateway node settings file `gateway-props.json` and install a gateway node to an existing logical gateway from the Management Portal.

Prerequisites:

- Check that your machine meets requirements. See Prerequisites to Install a Gateway Node.

To install a gateway node to an existing logical gateway:

1. From the **Gateways** tab, select the gateway in which you want to install the gateway node.

2. Click the  (Nodes) tab.
3. Configure the node properties file `gateway-props.json`.
   a. Click **Open Installation Wizard** to start configuring values for the `gateway-props.json` properties file.

      The first screen is displayed with information about your gateway.
   b. Click next **>** to continue.
   c. In the **Step 2: Node Properties Configuration** screen, complete required parameters marked with an asterisk (*). Complete optional parameters according to your environment. For a detailed description of all parameters, see gateway-props.json File.

      Required fields:

      - **Gateway Node Name**
      - **Listen IP Address**
      - **Publish Address**
      - **Node Installation Directory**
   d. In the **Step 3: Optional Additional Configuration** screen, complete any additional parameters that you would like to customize. For a detailed description of parameters, see gateway-props.json File.
   e. In the **Step 4: Download Properties File** screen, click **Download File** to download the file to the directory in which you extracted the Gateway Node Installer package.
4. Register the node to the logical gateway by running the `install-configure-start-join` installer action.

   This installs the gateway node, configures the domain, starts the node domain's servers, and registers the node to the gateway.

   Sample Linux command:

   ```
   ./APIGateway -f gateway-props.json -a install-configure-start-join
   ```

   When you run this action, you are prompted for the following user credentials:

   - **weblogic user**: the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.
   - **gateway manager user**: the gateway manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist and must be assigned the Administrator or Gateway Manager role. This user is issued the Manage Gateway grant when the gateway is created.
   - **gateway runtime user**: the gateway runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist and must be assigned the Gateway Runtime role. This user is issued the Node Service Account grant when the gateway is created.

You can view the log files for the installer actions here:

- `<nodeInstallDir>/logs`: contains log files for the installer actions
- `<nodeInstallDir>/GATEWAY_HOME/logs/wlst_<timestamp>.log`: contains WLST log files for the `configure` action

Where `<nodeInstallDir>` is the directory you installed the gateway into, specified by the `nodeInstallDir` property.

After the node is installed, approve the node's registration to the logical gateway. See Approve a Gateway Node Registration.

Installing Additional Gateway Nodes for a Logical Gateway

When you installed the first gateway node in the logical gateway, you generated the gateway node settings file `gateway-props.json` and downloaded the file to the directory in which you extracted the Gateway Node Installer package. Use the same `gateway-props.json` file to install additional gateway nodes.

Prerequisites

- Check that the node that you want to add to the logical gateway meets requirements. See Prerequisites to Install a Gateway Node
- Install the first gateway node in your logical gateway. See Installing the First Gateway Node for a Logical Gateway

To install additional gateway nodes in an existing logical gateway:

1. Register the node to the logical gateway by running the `install-configure-start-join` installer action and specifying the same `gateway-props.json` file that you used to install the first gateway node in the logical gateway.

   This installs the gateway node, configures the domain, starts the node domain's servers, and registers the node to the gateway.

   Sample Linux command:

   ```
   ./APIGateway -f gateway-props.json -a install-configure-start-join
   ```

   When you run this action, you are prompted for the following user credentials:

   - **weblogic user**: the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.
   - **gateway manager user**: the gateway manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist and must be assigned the Administrator or Gateway Manager role. This user is issued the Manage Gateway grant when the gateway is created.
   - **gateway runtime user**: the gateway runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist and must be assigned the Gateway Runtime role. This user is issued the Node Service Account grant when the gateway is created.

You can view the log files for the installer actions here:

- *<nodeInstallDir>*`/logs`: contains log files for the installer actions
- *<nodeInstallDir>*`/GATEWAY_HOME/logs/wlst_`*<timestamp>*`.log`: contains WLST log files for the `configure` action

Where *<nodeInstallDir>* is the directory you installed the gateway into, specified by the `nodeInstallDir` property.

After the node is installed, approve the node's registration to the logical gateway. See Approve a Gateway Node Registration.

Create a New Logical Gateway while Installing a Gateway Node

If you do not have a logical gateway created, you can install a gateway node and create a logical gateway at the same time.

Check that your machine meets requirements. See Prerequisites to Install a Gateway Node.

To create a new logical gateway and install a gateway node at the same time:

1. In the directory in which you extracted the Gateway Node Installer package, edit the `gateway-props.json` file with properties describing your API Platform Cloud Service instance and your gateway node.

   You must supply the following properties:

   - `nodeInstallDir`
   - `logicalGateway`
   - `gatewayNodeName`
   - `managementServiceURL`
   - `listenIpAddress`
   - `publishAddress`

   The following properties are not mandatory to install a gateway node, but may be required depending on your environment:

   - `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.
   - `nodeProxy`: required if the gateway node needs a proxy to pass client requests to backend services. You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.
   - `gatewayExecutionMode`: value of `Production` is required to enable SSL host name verification and certificate verification. If this property is not provided, it defaults to a value of `Development` and SSL host name verification and certificate verification are disabled.

   Remove any properties that you don't need and add any others your runtime environment requires. See gateway-props.json File.

2. Install the gateway node and create a logical gateway at the same time by running the `install-configure-start-create-join` installer action.

This installs the gateway node, configures the domain, starts the node domain's servers, creates a logical gateway, and registers the node to that gateway.

Sample Linux command:

```
./APIGateway -f gateway-props.json -a install-configure-start-create-join
```

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

You can view the log files for the installer actions here:

- `<nodeInstallDir>/logs`: contains log files for the installer actions
- `<nodeInstallDir>/GATEWAY_HOME/logs/wlst_<timestamp>.log`: contains WLST log files for the `configure` action

Where `<nodeInstallDir>` is the directory you installed the gateway into, specified by the `nodeInstallDir` property.

After the node is installed, approve the node's registration to the logical gateway. See Approve a Gateway Node Registration.

gateway-props.json File

The gateway installer zip includes a `gateway-props.json` file. Gateway installer actions use the values defined in this file. Edit the file to provide the values required for the installer actions you want to run.

> **Note:**
>
> The sample `gateway-props.json` file contains properties you may not need. Remove properties you don't need and their placeholder values. If the properties are still present in the file with placeholder values you may experience issues running actions.

See Gateway Node Installer Actions.

gateway-props.json Values

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
gatewayMServerPort	Managed Server Port	The HTTP Managed Server port of the gateway node. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `8011`.	`8011`	Optional
gatewayAdminServerPort	Administration Server Port	The HTTP Administration Server port of the gateway node. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `8001`.	`8001`	Optional
nodeInstallDir	Node Installation Directory	The directory where the gateway is installed or will be installed. **Note**: This directory must be different than the directory you unzipped the gateway installer into.	`/path/to/install`	Mandatory for all actions
prevInstallCleanupAction	-	This property indicates what should be done with a previous installation that may exist in the directory referred by `gatewayInstallDir` when any of the `install` actions are run. The currently supported options are `clean` (remove the contents of the `nodeInstallDir` before installing the gateway in that directory) and `archive` (move the contents of the `nodeInstallDir` to the location specified by the `installationArchiveLocation` property before installing the gateway). The default option is `clean`.	`clean`	Optional This property applies for only the following actions: • `install` • `install-configure` • `install-configure-start-create-join` • `install-configure-start-join` This property is not applicable for other actions.

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
installationArchiveLocation	Installation Archive Location	The directory where the archive of the current installation will be stored before a fresh install is initiated, or when the `archive` action is run. **Note**: This directory must be different than the directory you unzipped the gateway installer into.	`/path/to/archiveLocation`	Mandatory only if the `archive` option is specified in the `prevInstallCleanupAction` property. This property applies for only the following actions: • `install` • `install-configure` • `install-configure-start-create-join` • `install-configure-start-join` This property is not applicable for other actions.

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
logicalGateway	-	The name of the logical gateway created on the management service when running `create` actions.	`Production Gateway`	Mandatory for the following actions: • `create-join` • `creategateway` • `install-configure-start-create-join` This property is not applicable for other actions.
logicalGatewayId	Logical Gateway Id	The ID of the logical gateway the node registers to. This property must be supplied when registering to a logical gateway that already exists in the management tier.	`101`	Mandatory for the following actions: • `join` • `install-configure-start-join` This property is not applicable for other actions.

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
gatewayNodeName	Gateway Node Name	The name of the node gateway domain.	`gateway1`	Mandatory for the following actions: • `create-join` • `install-configure-start-create-join` • `install-configure-start-join` • `join` This property is not applicable for other actions.
gatewayAdminServerSSLPort	Administration Server SSL Port	The HTTPS Administration Server port of the gateway node. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `9027`.	`9027`	Optional
gatewayMServerSSLPort	Managed Server SSL Port	The HTTPS Managed Server port of the gateway node. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `9029`.	`9029`	Optional

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
managementServiceUrl **Note:** this property replaces the `managementServerHost` and `managementServerPort` properties from earlier releases.	Management Service URL	The URL of the management service instance you're registering the node to, in the following format: `https://<hostname or IP>:<port>`. If you have configured a hostname for API Platform Cloud Service, you should always use the hostname for this property. **Note**: To specify an HTTPS connection to the management service, you must prefix this property with `https` and use port 443. This should always be the case for provisioned service instances.	`https://example.com:443`	Mandatory for the following actions: • `create-join` • `creategateway` • `install-configure-start-create-join` • `install-configure-start-join` • `join` • `lockdown` • `reset` • `unregister` • `updatecredentials` • `updateoauthprofile` While not required to run the action, this property is used by the `status` action to determine the status of the connection to the management service. This property is not

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
				applicable for other actions.
oauthProfile Location	OAuth 2.0 Profile Location	This property refers to the local OAuth profile name used in context with the `updateoauthprofile` action.	`/path/to/OAuth2TokenLocalEnforcerConfig.xml`	Mandatory for the `updateoauthprofile` action. This property is not applicable for other actions.
listenIpAddress	Listen IP Address	The internal IP used for configuration of the gateway node domain. The value of this property should be a private IP address of the machine the node is installed to; this IP corresponds to the ethernet interface (eth0, eth1, etc.) over which client requests are received. Setting this property to `localhost`, `127.0.0.1`, or loopback IPs are not correct and may result in errors.	`192.0.2.0`	Mandatory for the following actions: • `configure` • `install` • `install-configure` • `install-configure-start-create-join` • `install-configure-start-join` • `lockdown` This property is not applicable for other actions.

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
publishAddress	Publish Address	The public IP address/hostname that is displayed in the management service for the node's URL. The node URLs in the UI are set to this address (suffixed by appropriate ports).	`gateway1.example.com`	Mandatory for the following actions: • `configure` • `install` • `install-configure` • `install-configure-start-create-join` • `install-configure-start-join` • `lockdown` This property is not applicable for other actions.
managementServiceConnectionProxy	Management Service Connection Proxy	A JSON array defining the HTTP/HTTPS proxies used by the gateway controller to pull down updates and deployments, provide acknowledgements, and send analytcs data. **Note:** HTTP or HTTPS in a proxy's URL refers to the URL the proxy uses; this doesn't necessarily indicate if a proxy is secured or not secured by SSL. If you don't need to use a proxy to reach the management service, make sure you: • remove this line from the properties file, or • set this property's value to an empty JSON array (like `[]`).	`["http://proxy.example.com:80","https://proxy.example.com:443"]`	Optional **Note**: this property is required at runtime if the gateway node needs a proxy to connect to the management service, as defined in the `managementServiceUrl` property.

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
nodeProxy	Gateway Node Proxy	A JSON array defining the HTTP/ HTTPS proxies used for outbound backend service calls. **Note:** HTTP or HTTPS in a proxy's URL refers to the URL the proxy uses; this doesn't necessarily indicate if a proxy is secured or not secured by SSL. If you don't need to use a proxy to reach your backend services, make sure you: • remove this line from the properties file, or • set this property's value to an empty JSON array (like `[]`). You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.	`["http://proxy.example.com:80","https://proxy.example.com:443"]`	Optional **Note**: this property is required at runtime if the gateway node needs a proxy to pass client requests to backend services.
coherencePort	Coherence Port	The Coherence port the gateway node domain uses. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `8088`.	`8088`	Optional
gatewayDBPort	Gateway Database Port	The Java DB port used by the gateway. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `1527`.	`1527`	Optional
dbHostName	-	The hostname by which the Java DB installed with the gateway is accessible. When the installer is run, this property is updated to use the value provided for the `listenIpAddress` property. **Note**: The only instance in which you should provide a value for this property is if the value of the `listenIpAddress` property has changed or the proper value was not provided when the installer was run initially.	`192.0.2.0`	Optional
nodeManagerPort	Node Manager Port	The node manager listen port. Provide this property when a port on the machine you are installing the node on to conflicts with the default value of `5556`.	`5556`	Optional

Property	Display Name in Wizard	Description	Example	Mandatory/ Optional
heapSizeGb	Heap Size (Gb)	The memory size (in GB) to be used for admin and managed servers. This value must be an integer. The default value is 2.	2	Optional
maximumHeapSizeGb	Maximum Heap Size (Gb)	Maximum memory size (in GB) allowed that can be used for admin and managed servers. This value must be an integer. The default value is 4.	4	Optional
gatewayExecutionMode	Gateway Execution Mode	Specifies the execution mode of the gateway node. Supported values are `Development` (default) and `Production`. When this property is set to `Development`, SSL hostname verification and certificate validation are turned off. These are enabled when this property is set to `Production`. If set to Production mode, ensure that the OTD public certificate is CA signed. See Obtaining a CA-Signed Certificate and Installing a Certificate in *Oracle Traffic Director Administrator's Guide* to import the certificate chain. In addition, ensure that the intermediate and root certificate of the CA-signed certificate installed on OTD is trusted by the trust store configured on the gateway. It is also recommended that the gateway should be configured with custom identity and custom trust or custom identity and Java standard trust. See Configure Keystores for WebLogic Server.	`Development` or `Production`	Optional
opatchesFolder	-	Specifies the location that contains patches you want to apply to the gateway node.	`/path/to/patches/folder`	Mandatory for the `applyPatches` action. This property is not applicable for other actions.

Example 3-1 Sample gateway-props.json File

```
{
   "nodeInstallDir" : "/path/to/install",
   "logicalGateway"       : "gateway1",
```

```
   "gatewayNodeName"   : "testGatewayNode",
   "managementServiceUrl"    : "https://example.com:443",
   "oauthProfileLocation"    : "/path/to/
OAuth2TokenLocalEnforcerConfig.xml"
   "listenIpAddress"              : "192.0.2.0",
   "publishAddress"      :  "gateway1.example.com",
   "managementServiceConnectionProxy" : ["http://proxy.example.com:
80",”https://proxy.example.com:443"],
   "nodeProxy"      : , "[http://proxy.example.com:80",”https://
proxy.example.com:443]"
   "gatewayExecutionMode": "Development",
}
```

Gateway Node Installer Actions

The gateway node installer supports multiple actions that you can perform on a gateway node.

Each action is executed by running the `APIGateway` gateway node installer and passing the action name in the `-a` or `--action` property. When executing an action, its mandatory and optional properties can be passed by adding them to the gateway-props.json File, passed with the `-f` or `--file` property, or passing them as key-value pairs using the `--keyvalue` or `-kv` properties (separate each pair with a space, like `–kv nodeInstallDir=<value> logicalGateway=<value>`).

Actions containing multiple hyphen-separated actions, like `install-configure-start-create-join`, perform all listed actions in sequence. Required properties for all actions must be provided by either including them in the gateway-props.json File passed with the `-f` or `--file` property, or passing them as key-value pairs.

> **Tip:**
>
> Run this command to view a full list of installer actions and options: `./APIGateway -h`

Topics

- applypatches
- configure
- create-join
- creategateway
- destroyNode
- install
- install-configure
- install-configure-start-create-join
- install-configure-start-join
- join

- lockdown
- reset
- start
- status
- stop
- unregister
- updatecredentials
- updateoauthprofile

applypatches

Patches an installed gateway node.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `opatchesFolder`

Example 3-2 Example patch Action

```
./APIGateway -f gateway-props.json -a patch
```

configure

Configures a gateway node domain.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `listenIpAddress`
- `publishAddress`

Optional Properties

This action also supports these optional properties:

- `heapSizeGb`
- `maximumHeapSizeGb`

Example 3-3 Example configure Action

```
./APIGateway -f gateway-props.json -a configure
```

create-join

Creates a new logical gateway with the management service and registers the gateway node to it.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `logicalGateway`
- `gatewayNodeName`
- `managementServerHost`
- `managementServerPort`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort`.
- `nodeProxy`: required if the gateway node needs a proxy to pass client requests to backend services. You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.

Example 3-4 Example create-join Action

```
./APIGateway -f gateway-props.json -a create-join
```

createGateway

Creates a new logical gateway with the management service.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `logicalGateway`
- `gatewayNodeName`
- `managementServerHost`
- `managementServerPort`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.

Example 3-5 Example createGateway Action

```
./APIGateway -f gateway-props.json -a creategateway
```

destroyNode

Unregisters the gateway node (which undeploys all APIs, applications, policies, and artifacts) and undeploys the gateway controller.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This

user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registered to.

> **Note:**
>
> To restore the node after running this action:
>
> - Run the stop action to stop the node's servers
> - Run the configure, start, and join actions in sequence to configure the node domain, start its servers, and then reregister it to a logical gateway on the management tier.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`

Example 3-6 Example destroyNode Action

```
./APIGateway -f gateway-props.json -a destroyNode
```

install

Installs a gateway node domain. You must also run the configure action to configure the domain and the start action to start the servers. Then you can either create and join a new logical gateway using the create-join action, or join an existing logical gateway.
When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `listenIpAddress`
- `publishAddress`

Example 3-7 Example install Action

```
./APIGateway -f gateway-props.json -a install
```

install-configure

Installs a gateway node and configures the domain.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. This user

is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `listenIpAddress`
- `publishAddress`

Optional Properties

This action also supports these optional properties:

- `heapSizeGb`
- `maximumHeapSizeGb`

Example 3-8 Example install-configure Action

```
./APIGateway -f gateway-props.json -a install-configure
```

install-configure-start-create-join

Installs and configures a gateway node domain, starts the domain's admin and managed server(s), creates a new logical gateway with the management service, and registers the node to it.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `logicalGateway`
- `gatewayNodeName`
- `managementServerHost`
- `managementServerPort`

- `listenIpAddress`
- `publishAddress`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.
- `nodeProxy`: required if the gateway node needs a proxy to pass client requests to backend services. You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.
- `gatewayExecutionMode`: value of `Production` is required to enable SSL hostname verification and certificate verification. If this property is not provided, it defaults to a value of `Development` and SSL hostname verification and certificate verification are disabled.

This action also supports these optional properties:

- `heapSizeGb`
- `maximumHeapSizeGb`

Example 3-9 Example install-configure-start-create-join Action

```
./APIGateway -f gateway-props.json -a install-configure-start-create-join
```

or

```
./APIGateway -a install-configure-start-create-join –kv
nodeInstallDir=<value> logicalGateway=<value> ...
```

install-configure-start-join

Installs and configures a gateway node domain, starts the domain's admin and managed server(s), and registers the node to an existing logical gateway with the management service.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. This user is created when you run this action. The user is stored in the gateway domain's local LDAP. When running other actions on this node, you must supply these credentials.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. This user must already exist on the Management Portal. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. This user must already exist on the Management Portal. This user is issued the Node Service Account grant when the gateway is created.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `logicalGateway`
- `gatewayNodeName`
- `managementServerHost`
- `managementServerPort`
- `listenIpAddress`
- `publishAddress`
- `logicalGatewayId`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.
- `nodeProxy`: required if the gateway node needs a proxy to pass client requests to backend services. You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.
- `gatewayExecutionMode`: value of `Production` is required to enable SSL hostname verification and certificate verification. If this property is not provided, it defaults to a value of `Development` and SSL hostname verification and certificate verification are disabled.

This action also supports these optional properties:

- `heapSizeGb`
- `maximumHeapSizeGb`

Example 3-10 Example install-configure-start-join Action

```
./APIGateway -f gateway-props.json -a install-configure-start-join
```

join

Registers the gateway node to an existing logical gateway on the management service.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This

user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registering to.

- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. This user must already exist on the Management Portal. This user must be issued the Node Service Account grant for the logical gateway the node is registering to.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `logicalGateway`
- `gatewayNodeName`
- `managementServerHost`
- `managementServerPort`
- `logicalGatewayId`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.
- `nodeProxy`: required if the gateway node needs a proxy to pass client requests to backend services. You can also provide a value for this property in the Management Portal UI. See Configuring a Proxy for a Gateway Node.

Example 3-11 Example join Action

```
./APIGateway -f gateway-props.json -a join
```

lockdown

Locks down the gateway node domain. See Gateway Node Lockdown.
When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `managementServerHost`
- `managementServerPort`
- `listenIpAddress`
- `publishAddress`

Optional Properties

If you need to use a proxy to reach the URLs defined in the `managementServerHost` or `managementServerPort` properties from the gateway node you must also provide values for the `managementServiceConnectionProxy` and `nodeProxy` properties.

Example 3-12 Example lockdown Action

```
./APIGateway -f gateway-props.json -a lockdown
```

reset

Resets the gateway node by fetching and redeploying all entities, like APIs, applications, policies, artifacts, and configurations deployed to the gateway node.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registered to.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `managementServerHost`
- `managementServerPort`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.

Example 3-13 Example reset Action

```
./APIGateway -f gateway-props.json -a reset
```

start

Starts the gateway node domain's servers. This action takes several minutes to complete.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`

This action assumes that the install and configure actions (or a compound action that performs all of these actions) have been completed successfully. The start action uses metadata generated from these actions and the `nodeInstallDir` property to identify the node to start.

Example 3-14 Example start Action

```
./APIGateway -f gateway-props.json -a start
```

status

Returns the results of all installer actions performed by a user, the status of the Management Tier and gateway node servers, and details about the gateway node domain environment.

See Viewing Gateway Node Status.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`

While the `status` action doesn't require any properties other than `nodeInstallDir`, it reuses properties that are required to be defined for actions that are run before `status`. For example, the status action does not require the `managementServerHost` and `managementServerPort` properties to run, it uses these to determine the status of the Management Service that a node is registered to.

Example 3-15 Example status Action

```
./APIGateway -f gateway-props.json -a status
```

stop

Stops the gateway node domain's servers. This executes asynchronously; it takes several minutes for the database and the servers to shut down completely.

When you run this action, you are prompted for the **weblogic user** credentials. These credentials belong to the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`

This action assumes that the install, configure, and start actions (or a compound action that performs all of these actions) have been completed successfully. The stop action uses metadata generated from these actions and the `nodeInstallDir` property to identify the node to stop.

Example 3-16 Example stop Action

```
./APIGateway -f gateway-props.json -a stop
```

unregister

Unregisters the gateway node from the specified logical gateway with the management service.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registered to.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `managementServerHost`
- `managementServerPort`
- `logicalGatewayId`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.

Example 3-17 Example unregister Action

```
./APIGateway -f gateway-props.json  -a unregister
```

updatecredentials

Updates the Gateway Runtime user credentials used by the node. The Gateway Runtime user is used to poll for updates to the logical gateway node definition and to send analytics data to the management tier. Perform this operation after updating the user's credentials on the management tier.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registering to.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the gateway. This user must already exist on the Management Portal. This user must be issued the Node Service Account grant for the logical gateway the node is registering to.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `managementServerHost`
- `managementServerPort`

Optional Properties

The following properties are not mandatory to run this action, but may be required (if you have not already defined them) depending on your environment:

- `managementServiceConnectionProxy`: required if the gateway node needs a proxy to connect to the management service, as defined in the `managementServerHost` or `managementServerPort` properties.

Example 3-18 Example updatecredentials Action

```
./APIGateway -f gateway-props.json -a updatecredentials
```

updateoauthprofile

Updates the OAuth profile of the gateway node. The action reads the file specified by the `oauthProfileLocation` property and updates the gateway node OAuth profile accordingly.

When you run this action, you are prompted for the following user credentials:

- **weblogic user**: the WebLogic administrator user of the gateway node. These credentials are created when you run any of the `install` actions.
- **gateway manager user**: the Gateway Manager user that is responsible for managing this gateway. You must provide the user's name and password. This user must already exist on the Management Portal. This user must be issued the Manage Gateway grant for the logical gateway the node is registered to.

Required Properties

This action requires that the following properties are defined in gateway-props.json File or passed when the action is run as key/value pairs:

- `nodeInstallDir`
- `managementServerHost`

- `managementServerPort`
- `oauthProfileLocation`

Example 3-19 Example updateoauthprofile Action

```
./APIGateway -f gateway-props.json -a updateoauthprofile
```

Updating Gateway Node Properties

You update gateway node properties with the gateway node installer. Edit the `gateway-props.json` file, only include properties that you want to update, and run the actions that correspond to the updated properties.

To update gateway node properties:

1. Identify which properties you want to change and which actions you need to run to update desired properties.

   For a list of properties and the corresponding actions, see gateway-props.json File.

2. Edit the `gateway-props.json` file, add properties you want to update, and remove any properties you don't want to update.
3. Run the actions that correspond to the changed properties.

   For a detailed description of each installer action with examples, see Gateway Node Installer Actions.

4. Run the installer actions with the same `gateway-props.json` file on each gateway node that you want to update.

Viewing Gateway Node Status

You can view gateway node installer action results, Management Server and gateway node server status, and gateway node domain environment details with the `status` installer action.

What You See When Viewing Gateway Node Status

The following information is returned when you run the `status` gateway node installer action:

- **Actions**: lists the gateway node installer actions the user has performed, including the result: `SUCCESSFUL` if the action succeeded, `not_attempted` if the action was not attempted, and `FAILED` if the action failed. If applicable, additional details may be returned for an action. For example, if the `patches` action was attempted, details about successfully and unsuccessfully applied patches are displayed.
- **Servers**: returns lists the status of the Management Tier server in the node's `gateway-props.json` file and the gateway node domain itself. A REST call is sent to the endpoint that returns the server's version. If defined, the request to the Management Server is sent through the `phoneHomeProxy` listed in the `gateway-props.json` file or passed as a key-value pair when running the action. If successful, the server is listed as `running`. If unsuccessful, the server is listed as `not_accessible`.

- **Environment details**: lists the free space available (in bytes) available to the gateway node's `domain` directory.

To view a gateway node's status:

1. Sign in to the machine the gateway node is installed on.
2. Navigate to the directory the gateway node installer was extracted to: `cd /path/to/installer`
3. Run the status action: `./APIGateway -f gateway-props.json -a status`

   You are prompted for the **weblogic** user's credentials. These are the WebLogic administrator user credentials that were supplied when the gateway node was installed. This user is stored in the gateway node domain's local LDAP.

The gateway node status is displayed.

Configuring Gateway Node Domains

Configure the domains for each of your gateway nodes, including configuring authentication providers, SSL certificates for passing requests to HTTPS endpoints, and locking down your nodes.

Topics

Signing into the WebLogic Adminstration Console for a Gateway Node Domain

Sign in to the WebLogic Server Administration Console for your gateway node domain to configure the WebLogic Server authentication providers.

To sign into the WebLogic Administration Console:

1. Open a browser window and navigate to: `http://<hostname>:<port>/console`

   Where <*hostname*> is the DNS name or IP address of the Administration Server and <*port*> is the address of the port on which the gateway node Administration Server is listening for requests (8001 by default).

> **Note:**
>
> If your browser is configured to send HTTP requests to a proxy server, you may need to configure your browser so that it does not send Administration Server HTTP requests to the proxy. If the Administration Server is running on the same machine as your Web browser, configure your browser so that requests sent to `localhost` or IP address `127.0.0.1` are not sent to the proxy server.

2. When the login page appears, enter the user name and the password you used to start the Administration Server (you may have specified this user name and password during the installation process), or enter a user name that is granted one of the default global security roles.

Supported WebLogic Authentication Providers

Oracle API Platform Cloud Service - Classic supports WebLogic authentication providers in the gateway node domain for authenticating users in identity management systems.

Name	Description
WebLogic Authentication provider	Accesses user and group information in WebLogic Server's embedded LDAP server.
Oracle Internet Directory Authentication provider	Accesses users and groups in Oracle Internet Directory, an LDAP version 3 directory.
Oracle Virtual Directory Authentication provider	Accesses users and groups in Oracle Virtual Directory, an LDAP version 3 enabled service.
LDAP Authentication providers	Access external LDAP stores. You can use an LDAP Authentication provider to access any LDAP server. WebLogic Server provides LDAP Authentication providers already configured for Open LDAP, Sun iPlanet, Microsoft Active Directory, and Novell NDS LDAP servers.
RDBMS Authentication providers	Access external relational databases. WebLogic Server provides three RDBMS Authentication providers: SQL Authenticator, Read-only SQL Authenticator, and Custom RDBMS Authenticator.
WebLogic Identity Assertion provider	Validates X.509 and IIOP-CSIv2 tokens and optionally can use a user name mapper to map that token to a user in a WebLogic Server security realm.
SAML Authentication provider	Authenticates users based on Security Assertion Markup Language 1.1 (SAML) assertions
Negotiate Identity Assertion provider	Uses Simple and Protected Negotiate (SPNEGO) tokens to obtain Kerberos tokens, validates the Kerberos tokens, and maps Kerberos tokens to WebLogic users

Name	Description
SAML Identity Assertion provider	Acts as a consumer of SAML security assertions. This enables WebLogic Server to act as a SAML destination site and supports using SAML for single sign-on.

See also About Configuring the Authentication Providers in WebLogic Server in *Administering Security for Oracle WebLogic Server*.

Configure WebLogic Authentication Providers

You must configure an authentication provider for your identity management system for gateway nodes to authenticate which users can send requests to APIs secured with Basic Auth and OAuth 2 policies.

You perform many of these actions in the WebLogic Server Administration Console of your gateway node domain.

To configure WebLogic authentication providers:

1. Access the Weblogic Administration Console in your browser by navigating to `http://<hostname>:<port>/console` where `<hostname>` is the DNS name or IP address of the Administration Server and `<port>` is the address of the port on which the gateway node Administration Server is listening for requests (8001 by default).
2. See to the relevant topic in *Administering Security for Oracle WebLogic Server* for the authentication provider you want to configure for your identity domain:
   - Configuring the WebLogic Authentication Provider
   - Configuring LDAP Authentication Providers
   - Configuring RDBMS Authentication Providers
   - Configuring the Windows NT Authentication Provider
   - Configuring the SAML Authentication Provider
   - Configuring the Password Validation Provider
   - Configuring Identity Assertion Providers
   - Configuring the Virtual User Authentication Provider
3. **(Optional)** Complete these steps only if you are replacing the embedded LDAP with the identity provider you are configuring:
   a. Map the appropriate Admin role in your identity provider to the WebLogic **Administrators** role, available in the WebLogic XACML role mapping provider.
   b. In the `boot.properties` file, replace the encrypted username and password in the file with the username and password (in plain text) of the new admin user.

      The server encrypts these values when the server starts.
   c. Remove the default authenticator. See Delete security providers in *Oracle WebLogic Server Administration Console Online Help*.

4. Restart the gateway WebLogic Server domain. See Starting and Stopping Servers in *Administering Server Startup and Shutdown for Oracle WebLogic Server 12.1.3*.

Configure SSL Certificates to Pass Requests to Services Over HTTPS

To pass requests to backend service endpoints using the HTTPS protocol, you must first import the required SSL certificates into the WebLogic trust stores for your gateway node domains.

1. Navigate to the following directory on the machine from which your gateway node is running:

   ${GW_INSTALL_DIR}/GATEWAY_HOME/wlserver/server/lib
   Where *${GW_INSTALL_DIR}* is the directory where you installed the gateway node.

2. To import the external CA into the WebLogic trust store, execute the following command:

   ```
   keytool -import -alias myCa1 -trustcacerts -file ${CA_FILE_NAME} -
   keystore <trust keystore> -storepass <trust keystores Passphrase>
   DemoTrustKeyStorePassPhrase
   ```

   Where *myCa1* is the keystore alias, *${CA_FILE_NAME}* is the CA file you want to import, *<trust keystore>* is the name of the keystore, and *<trust keystores Passphrase>* is the keystore's passphrase.

   The command depends on the Keystore type that you selected WebLogic console at `servers, managedServer1, Configuration, Keystores`. Change the command depending on the CA certificate that you inserted for the gateway. `Keystores` page provides you 4 choice:

   - Custom Identity and Command Line Trust (not supported)
   - Custom Identity and Custom Trust
   - Custom Identity and Java Standard Trust
   - Demo Identity and Demo Trust (default option)

   <trust keystore> value is found at `Keystores` page.

3. Repeat this procedure for each gateway node that needs to pass requests to the same back-end service endpoints over HTTPS.

Gateway Node Lockdown

A Gateway Node exposes REST endpoints for the APIs deployed on it. Apart from the deployed APIs, the nodes have internal REST endpoints which need to be secured. Lockdown will restrict access to the internal endpoints to just the local servers in the domain.

Topics

- Endpoints on a Gateway Node
- Lock Down a Gateway Node
- Additional Gateway Node Lockdown Scenarios

Endpoints on a Gateway Node

Learn about the endpoints exposed by gateway nodes.

Gateway nodes expose the following types of REST endpoints:

- Gateway controller endpoints
- Internal endpoints
- Weblogic Administration Console endpoint
- Deployed API Endpoints

Gateway controller endpoints function as management control for the gateway node, allowing the user to control the nature of polling and state of the node. The endpoints on the gateway node use the **/apiplatform** context root and are deployed on the managed servers. See REST API for the Gateway Controller in Oracle API Platform Cloud Service.

The gateway node exposes internal REST endpoints for the gateway controller to invoke to be able to deploy APIs and applications. These internal endpoints uses the **/prm_pm_rest** context root and are deployed on the node domain's managed servers.

The Weblogic Administration Console endpoint is deployed on the Administration Server of the gateway node domain. This endpoint uses **/console** context root only on the Administration Server.

All deployed Oracle API Platform Cloud Service - Classic APIs will have endpoints on the managed servers. The context roots will be decided by the API Manager. API Managers can't use **/prm_pm_rest** and **/apiplatform** as API endpoints.

Lock Down a Gateway Node

Use the `lockdown` gateway node installer action to lockdown a node.

1. Before you run the `lockdown` action, ensure that you've added the internal IP address of the machine onto which the node is installed in the `listenIpAddress` property in gateway-props.json File.

   > **WARNING:**
   >
   > Failing to add the internal IP address of the machine to the `listenIpAddress` property before running the `lockdown` action will break operation of the gateway controller.

2. Ensure that you have gateway manager runtime user credentials.
3. Run the `lockdown` installer action.

All REST calls to locked endpoints (with `/prm_pm_rest` and `/apiplatform` contexts) from any host other than the machine you specified are rejected with the following HTTP 403 Forbidden error: `IP address is not allowed`.
To undo the lockdown, replace the IP values of the `listenIpAddress` with `*` (an asterisk) and run the `lockdown` action again.

Additional Gateway Node Lockdown Scenarios

Learn about additional gateway node lockdown scenarios.

Locking Down the Administration Server

The WebLogic Administration Console is accessible only using the administrative user credentials for the gateway node domain. This user is created when you install a gateway node domain. As this user, you can shut down the Administration Server to further secure the gateway node. The gateway node performs in limited capacity. After shutting down the Administration Server, the following gateway controller endpoints are unavailable:

- `/apiplatform/gatewaynode/v1/security/credentials`
- `/apiplatform/gatewaynode/v1/security/profile`
- `/apiplatform/gatewaynode/v1/registration`
- `/apiplatform/gatewaynode/v1/registration`

See REST API for the Gateway Controller in Oracle API Platform Cloud Service.

Multiple Ethernet Interfaces Exist

If multiple ethernet interfaces exist for the machine you install a node to, the `lockdown` action uses the ones specified in the `listenIpAddress` property in gateway-props.json File. Loopback IPs or use of `localhost` is not supported by the `lockdown` action.

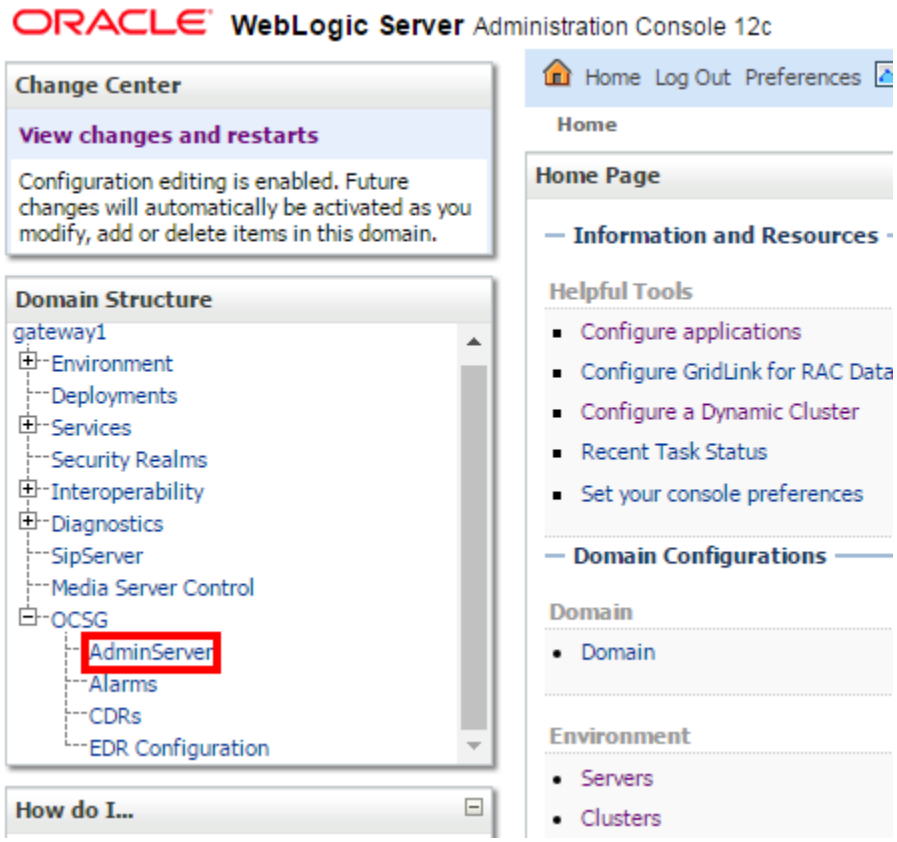
> **Note:**
>
> Loopback IPs or `localhost` invocations to the gateway do not work and are not supported, even before lockdown.

Configure Gateway Node Firewall Properties in the WebLogic Adminsitration Console

Gateway nodes allow firewall properties to be configured. These properties will apply to all incoming traffic of `application/json` and `application/xml` types to APIs deployed on the gateway.
These limitations are not applied on the management endpoints that gateway controller exposes.

You can also configure firewall properties for all nodes registered to a gateway in the Management Portal.

1. Sign in to the node domain's WebLogic Server Administration Console as a user with administrative privileges.
2. From the Domain Structure panel, expand **OCSG**, and then click **AdminServer**.



3. On the Oracle Communication Services Gatekeeper page, expand **Container Services**, and then click **ApiFirewall**.
4. Update any of the firewall properties you want to change:
   - **MaxMessageSize**: Specifies the maximum size, in bytes, of the request, excluding attachments. The default value is set to 1024000. The maximum allowed value is 200MB.
   - **MaxUnboundedItems**: Specifies the maximum number of unbounded items that a message can contain. The default value is set to 1024.
   - **MaxItemValueLength**: Specifies the maximum size of a single message entity, such as an element, attribute or comment. The default value is 102400.
   - **MaxChildElementDepth**: Specifies the maximum number of nested elements allowed in a message. The default value is 1024 nested elements.
5. Click **Update Attributes** to save your changes.

The firewall properties you specified or updated are now enforced. You don't need to restart the gateway node domain.

Additional Firewall Properties

Your network implementation can be vulnerable to denial of service (DOS) attacks, which generally try to interfere with legitimate communication inside the Gateway in Oracle API Platform Cloud Service - Classic.

To prevent these messages from reaching your network, Gateways in Oracle API Platform Cloud Service - Classic offer configurable RESTful message filtering. You configure this filtering behavior by using the ApiFirewall configuration MBean. ApiFirewall determines how Gateways in Oracle API Platform Cloud Service - Classic filters messages attempting to enter Oracle API Platform Cloud Service.

Attack Strategy	Protection Strategy	Default Result
Malicious Content Attack, including: RESTful message attacks: • Oversize message layouts. • Oversize JSON or element values. • Oversize JSON array elements. • Messages with an inordinately large number of nested elements.	The ApiFirewall MBean settings (application tier) limit the acceptance of oversize message entities.	Rejects the message and returns the error message specified with the ErrorStatus attribute of ApiFirewallMBean.
Continuous wrong password attack.	The default WebLogic Security Provider setting (application tier) locks a subscriber out for 30 minutes after 5 wrong password attempts. This behavior is configurable. See the section on Protecting user Accounts in *Administering Security for Oracle WebLogic Server* for more information.	Rejects the message and returns a 500 Internal Server Error message.
External Entity Reference	Gateways in Oracle API Platform Cloud Service ApiFirewall (application tier) prohibits all references to external entities. It is possible to remove this protection.	Rejects the message and returns a 500 Internal Server Error message.

Configure Analytics Properties

Oracle API Platform Cloud Service - Classic gateway nodes use Logstash to collect analytics. Logstash aggregates data on each node before sending it to the management tier. Configure properties on each gateway node to determine how analytics are collected and how data is sent to the management tier.

To configure analytics properties:

Ensure that you perform this task as the user who installed the gateway. That user owns the `analyticsagent.properties` file you edit.

1. In a plain text editor, open the `analyticsagent.properties` file in this directory on the gateway node domain: *<GATEWAY_DOMAIN_HOME>*`/apics/analytics/`.

2. Edit the properties you want to change, and then save the file.

Property Name	Description	Default Value
`excluded_fields`	Specifies which fields are excluded from generated EDR log file. Setting to `None` prints all fields. Separate multiple fields with commas. Valid values: `None`, `Class`, `Method`, `ServiceName`, `ServiceProviderGroup`, `ServiceProviderId`, `Source`, `TagEdr`, `Direction`, `TsAfNT`, `TsBeNT`, and `status`.	`Class,Source,TagEdr,Direction,AccessUrl,AppInstanceId,ServiceName,ServiceProviderGroup,ServiceProviderId,TransactionId,TsAfNT,TsBeNT,ErrCat,status`
`rotate_filesize`	Size limit of log files. Rotation is timer based, so size will not be precise. Example values include: `100k`, `10m`, `2g`	`500M`
`max_logs`	Number of log files to retain. Once this value is reached, older archives are deleted on subsequent rollovers. Value of `0` means all log files are retained and none are deleted.	`10`
`edr_buffersize`	Size of the EDR log buffer. Example values include: `100k`, `10m`, `2g`	`4m`
`edr_flush_interval`	EDR log flush interval, in milliseconds.	`5000`
`logstash_check_interval`	Logstash process health check interval. If Logstash process is not detected, Analytics Agent attempts to restart. Example values include: `100ms`, `30s`, `5m`, and `1h`.	`15s`

Property Name	Description	Default Value
`stop_retry_logstash_after`	Time limit for Logstash restart attempts, after which Agent will no longer try to restart Logstash. Run *<GATEWAY_DOMAIN_HOME>*`/apics/analytics/manageLogstash.sh start` to reset this state. Setting this property to 0 means there is no time limit; the Agent will never stop trying to restart Logstash. Example values include: `0`, `100ms`, `30s`, `5m`, and `1h`.	`5m`
`logstash_install_dir`	Logstash installation directory, specified as a path relative to `domain/gateway1`. Logstash is installed at the default value when you install a gateway node domain. **Note**: You should not change this value. It is the only supported Logstash install location.	`../../install/LOGSTASH_HOME/logstash-2.2.2`
`logstash_worker`	Number of Logstash workers. Default value is the number of cores.	-
`logstash_pipeline_batch_size`	Maximum number of events an individual worker thread collects before attempting to execute filters and outputs. Larger batch sizes are generally more efficient, but increase memory overhead.	`125`
`logstash_heap_size`	Logstash heap size. Sets LS_HEAP_SIZE environment variable when starting Logstash.	-
`logstash_pipeline_batch_delay`	Maximum amount of time, in milliseconds, that Logstash waits for new messages after receiving an event in the current pipeline worker thread. This option adjusts the latency of the Logstash pipeline. This rarely needs to be tuned.	`5`
`logstash_custom_params`	Custom Logstash parameters, appended at end of the command line invoking logstash.	`--verbose -l /tmp/logstash.out`

Property Name	Description	Default Value
`apics_analytics_latency_min`	Minimum time, in milliseconds, between statistics uploads to the management tier.	`5000`
`apics_analytics_map_expiration`	Map expiration, in milliseconds of processing time, per time period. This value must be longer than the amount of time it takes to process one time period. A use case for increasing this value is to handle nonsequential processing of multiple days of logs spread over a large number of log files.	`1800000`
`apics_analytics_log_replay_limit`	Maximum number of time periods to replay, starting from most recent, on Logstash restart. Smaller numbers help restart performance. Larger numbers provide data reliability on restart when processing partially sorted log files. Empty time periods are not counted. Out of order data belonging to time periods before the start of replay are lost.	`10`
`apics_analytics_insecure_ssl`	Allow transmission of statistics to the management tier over insecure SSL. Transmission over insecure SSL is disabled when this value is `false`. Valid values include: `false`, `logstash`, and `curl`.	`false`
`always_reconfig_restart_logstash`	Logstash conf/* files will be recreated from templates/* on Logstash restart if set to `true`.	`false`

The configuration is checked about every 10 or 15 seconds. When changes are detected, Logstash restarts and the updated properties take effect. You don't need to restart the node domain.

About Logstash Retry Logs

The Oracle API Platform Cloud Service - Classic uses retry logs with logstash to improve performance and reliability. If the network has a high load, bad connectivity, or

goes up and down frequently, the retry logs prevents logstash from stopping and starting, which causes performance issues on the gateway.

In the case of processing failures in which a CSV cannot be sent or an EDR cannot be processed, events are written to separate log files which then feeds back into logstash. Multiple files prevent concurrency issues, since only one file can be written at a time. Each type of failure has its own log file:

- When EDRs are consumed before logstash has been configured, the individual EDR log lines that failed are written to a `Logs/RETRY_EDR_YYYY-MM-dd-HH-mm.%{edr_log_basename}.log` file. Since tenant credentials can be updated dynamically, the `config json` file can be sent at any time while logstash is running. One effect of this dynamic update is that EDRs can be read before the configuration file is sent. In these cases, these early EDRs are sent back into the `RETRY_EDR` log, and picked up by a separate logstash file Input, with a slower poll interval. These EDRs are aggregated by log_file matching the `RETRY_EDR` name.
- When the management tier does not send an HTTP 200 to the plugin because of bad credentials, HTTP timeouts, or other server errors, the retry events are written to a `Logs/RETRY_HTTP_YYYY-MM-dd-HH-mm.log` file. The CSV in these `RETRY_HTTP` events is still aggregated by the original log file name.
- When there is an HTTP failure if the tenant credentials are missing, the retry events are written to a `Logs/RETRY_CRED_YYYY-MM-dd-HH-mm.log` file. The processing behavior is the same as the `RETRY_HTTP` logs. They go into a separate log to avoid concurrency issues. This type of retry should not happen in single tenant mode.

  In multi-tenant mode, it can happen between the time of tenant onboarding and the time logstash gets the updated configuration file containing the new credentials.

In some rare cases, it is possible for EDR totals in the database not to match the totals in the log files. This is possible when logstash failure overlaps with management tier failure. The mismatch should be relatively small.

Every retry creates a new line in one of the `RETRY` log files. If logstash is restarted, it may try to reprocess the RETRY log files. Since the logic uses `?increaseOnly=true`, no data is lost or overwritten.

Retry log files should be purged by file age and `sincedb` status. Purging is done by the Analytics agent.

Enable Analytics in Production Environments

After provisioning an Oracle API Platform Cloud Service - Classic instance and configuring your gateway node domains, you must enable analytics for each gateway node domain.

To configure analytics in production environments:

In the `analyticsagent.properties` file in a gateway node domain, set the `apics_analytics_insecure_ssl` property value to `logstash`. Your changes take effect after about 10–15 seconds; Logstash automatically restarts on your node domain and analytics are enabled.

Repeat this task for each gateway node domain registered with your Oracle API Platform Cloud Service - Classic instance.

Managing Gateway Settings

View and manage logical gateway properties, including firewall settings for all nodes registered to the gateway.

Topics

Understanding the Gateway List Page

The Gateways List page displays all logical gateways created in the Management Portal.

A description, load balancer URLs, the number of APIs deployed to the gateway, and the number of requests (API deployment requests or node registration requests) that need attention are displayed for each logical gateway.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.
- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even in words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

  > **Note:**
  >
  > If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Viewing Gateway Details

You can view the details of a gateway in a side panel available from any of the tabs.

The side panel displays the following details:

- The name of the gateway.
- The description of the gateway.
- The most recent date and time that changes were saved for the gateway, and name of the user who saved them.

To view gateway details:

1. On the Gateways List page, select the gateway for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the gateway.

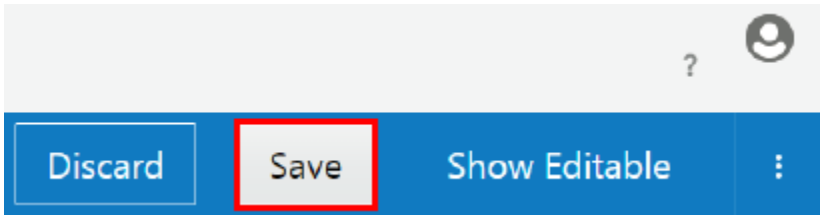
Editing Gateway Details

You can edit the name and description of a gateway in the Management Portal.

To edit the name or description of a gateway:

1. On the Gateways List page, click the gateway you want to edit.
2. Click the drawer icon to display the side panel.

3. Click the gateway name to edit it.
4. Click the description to edit it.
5. Click **Save**.



Configuring Gateway Firewall Properties

Configure gateway firewall properties in the Management Portal. These properties apply to all nodes registered to a gateway.

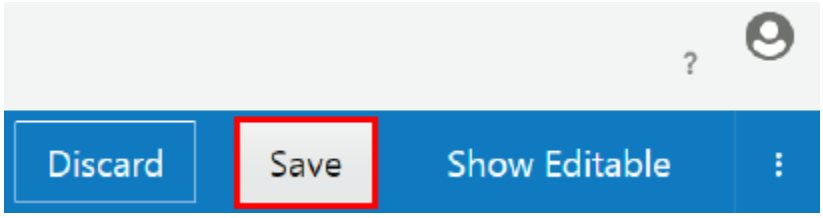
The gateway rejects messages that exceed any of the properties you set with a `400 Bad Request` HTTP response.

You can also configure a gateway node's firewall properties in its WebLogic Administration Console, as described in Configure Gateway Node Firewall Properties in the WebLogic Adminsitration Console. Note that properties you set in a node's WebLogic Server Administration Console apply for only that node.

Gateway Managers must be issued the Manage Gateway grant for a gateway to configure its properties.

To configure gateway properties:

1. From the Gateways List page, click the gateway for which you want to configure firewall properties.
2. On the **Settings** tab, update any of the firewall properties that you want to change:
   - **Maximum Message Size**: Specifies the maximum size, in bytes, of the request, excluding attachments. The default value is set to 1024000. The maximum allowed value is 200MB.
   - **Maximum Number of Unbounded Items**: Specifies the maximum number of unbounded items that a message can contain. The default value is set to 1024.
   - **Maximum Size of a Single Message Entry**: Specifies the maximum size of a single message entity, such as an element, attribute or comment. The default value is 102400.
   - **Maximum Nested Elements in a Message**: Specifies the maximum number of nested elements allowed in a message. The default value is 1024 nested elements.
3. Click **Save**.



The properties you configured are applied immediately. You don't have to restart gateway node domains to enable them.

Manage Gateway Nodes in the API Platform Cloud Service Management Portal

Manage registration, polling intervals, and proxies for each of your gateway nodes in the Oracle API Platform Cloud Service - Classic Management Portal.

Topics

- Unregister a Gateway Node

Understand Gateway Node Details

The Nodes page displays details for all nodes known to a gateway.

The Active tab displays all active gateway nodes. The name, description, and host is shown for each gateway.

The Requesting Registration tab displays gateway nodes that are requesting registration but have not yet been approved. The name, description, and host is shown for each node. The name of the user requesting the registration and the time the request was made are also displayed.

The Rejected tab displays gateway nodes that have requested registration, but have been rejected.

> **Tip:**
>
> Click **Dismiss** to clear a gateway node from the Rejected tab.

Register a Node to a Logical Gateway

Use the `join` node installer action to register a node to an existing logical gateway on the Management Portal. If you want to create a new logical gateway and register a node to it, use the `create-join` action instead.

> **Tip:**
>
> You can also register a node to a logical gateway with the REST API for the Management Service in Oracle API Platform Cloud Service.

To register a node to a logical gateway:

1. Edit or add the following properties in the `gateway-props.json` file with details describing the gateway and the management tier:
   - `logicalGateway`
   - `managementServerHost`
   - `managementServerPort`
   - **(Required for create-join actions only)**: `publishAddress`

   See gateway-props.json File.
2. Run the `join` action to register a node to an existing logical gateway, or run the `join` action to create a new logical gateway and register a node to it

   When you run this action, you are prompted for the following user credentials:
   - **weblogic user**: the WebLogic administrator user of the gateway node.

- **gateway manager**: the Gateway Manager user that is responsible for managing this gateway. This user is issued the Manage Gateway grant when the gateway is created.
- **gateway runtime user**: the Gateway Runtime user that is used to download configuration from and upload statistics to the API Platform Cloud Service Management Portal. This user is issued the Node Service Account grant when the gateway is created.

See join and create-join.

3. Sign in to the Management Portal with the Gateway Manager user specified in the previous step and approve the node registration.

The node is registered to the logical gateway on the Management Portal. The Gateway Manager user you specified is issued the Manage Gateway grant for the gateway and the Gateway Runtime user you specified is issued the Node Service Account for the gateway.

Approve a Gateway Node Registration

A gateway manager or administrator issued the Manage Gateway grant must approve node registrations before APIs can be deployed to the gateway. Approve a gateway registration in the Management Portal. Gateway Managers must be issued the Manage Gateway grant for a gateway to approve node registrations to it.

To approve a gateway registration:

1. From the Gateways List page, click the gateway for which you want to approve node registrations
2. Click the  (Nodes) tab.
3. Click the **Requesting** tab.
4. Hover over the gateway node registration you want to approve, and then click **Approve**. Click **Reject** to reject the registration instead. You can see rejected requests on the Rejected tab.

The registration is approved. You can now manage the gateway node in the Management Portal.

Changing the Node Polling Interval

Configure how regularly a gateway node polls the management tier for the logical gateway definition.

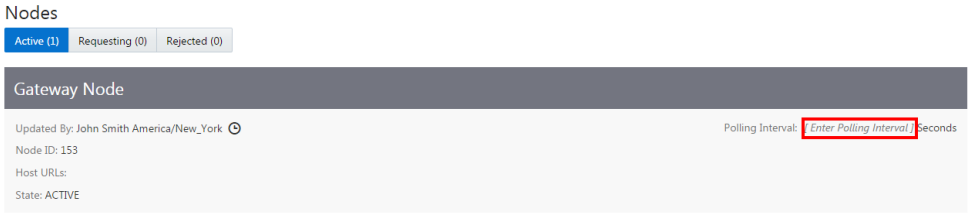
When a node polls the management tier and the definition for the gateway it is registered to differs from its current shape, the node updates itself to match the definition from the management tier. See Understanding Gateways and Gateway Nodes for an explanation of gateways and gateway nodes.

Gateway Managers must be issued the Manage Gateway grant for a gateway to change the polling interval of a node registered to it.

> **Note:**
>
> Different nodes belonging to a gateway can have different polling intervals. Nodes registered to the same gateway may not have the same endpoints or API iterations deployed, depending on the delay between polls from each node.

1. From the **Gateways** page, click the gateway the node is registered to.
2. Click the  (Nodes) tab.
3. Click **[Enter Polling Interval]**, or the interval itself, if one was already configured, and then enter polling interval as a positive integer, and then press **Enter**.

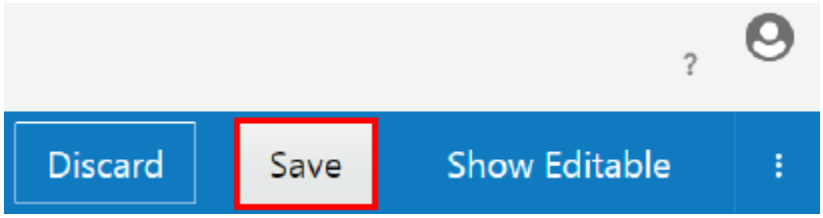
> **Note:**
>
> Entering `0` stops polling on this node.



4. Click **Seconds**, or other time period if one was already configured, and then click again to expand the time period list. Select the time period (**Seconds**, **Minutes**, **Hours**, or **Days**) you want the node to use to poll for updates.

For example, if you entered `5` as the polling interval and **Minutes** as the time period, the node polls for changes to the logical gateway definition every five minutes. If the deployed endpoints or API implementations differ, the node is updated to match the definition of the gateway from the Management Portal.

5. Click **Save**.



The polling interval is configured. The node will poll the management tier for the gateway the next time the previous interval elapses, and then the new polling interval takes effect.

> **Note:**
>
> If the previous polling interval was `0`, you must manually initiate a poll to synchronize the node with the latest logical gateway definition. The poll will not happen automatically.

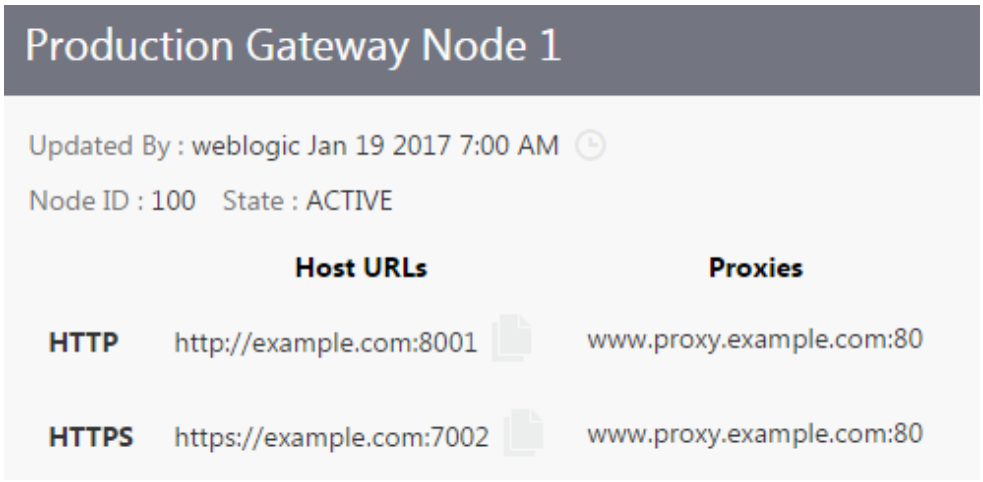
Configuring a Proxy for a Gateway Node

Configure proxies for each node registered to gateways you manage. This allows for different proxy configurations for development and production gateways or different proxies for nodes located in separate data centers. When configured, the node passes requests to your backend services through the proxies you specify.

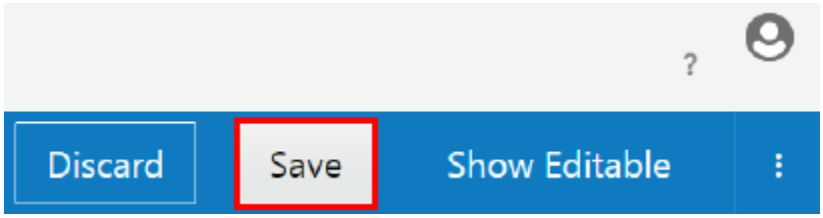
Gateway Managers must be issued the Manage Gateway grant for a gateway to configure proxies for nodes registered to it.

To configure a proxy for a gateway node:

1. In the Management Portal, click the **Gateways** tab.
2. Click the gateway the node is registered to.
3. Click the  (Nodes) tab.
4. In the **Proxies** column, enter the host and port of the HTTP and HTTPS proxies you want to pass requests through. For example, enter `www.proxy.example.com:80` in the HTTP and HTTPS rows.



5. Click **Save**.



The gateway node proxy is configured. You don't need to restart the node for this change to take effect.

Unregister a Gateway Node

Unregister a node from a gateway to stop managing it with the Oracle API Platform Cloud Service - Classic management tier.

Unregistering a node does not shut it down; it continues to run normally until you manually shut it down using the stop node installer action.

When you unregister a gateway, the node stops polling the management server for updated logical gateway definitions. As a result, new APIs or more recent API iterations deployed to the gateway will not be deployed to the unregistered node; however, APIs that were deployed to the node when it was unregistered are still deployed. Requests sent to these APIs are handled normally.

> **Tip:**
>
> You can also unregister a node using the REST API for the Management Service in Oracle API Platform Cloud Service or the unregister node installer action.

Gateway Managers must be issued the Manage Gateway grant for a gateway to unregister nodes from it.

To unregister a gateway node from the Management Portal:

1. From the Gateways list page, click the gateway the node is registered to.
2. Click the  **Nodes** tab.
3. Hover over the node you want to unregister, and then click **Unregister** when it appears.

> **Tip:**
>
> Instead of hovering, you can also click anywhere inside of or use the keyboard to navigate to the node panel to make the **Unregister** button appear.

4. Click **Yes** to confirm.

The gateway node is unregistered.
Run the stop node installer action to stop the running node server. Run the destroyNode node installer action to stop the running server and uninstall the node from the machine.

To reregister a node to a gateway, use the REST API for the Management Service in Oracle API Platform Cloud Service or the join gateway installer action.

Managing Gateway Grants

Gateway grants allow you to issue fine-grained permissions to users or groups for specific gateways.

Topics

Understanding Gateway Grants

Gateway grants are issued per gateway.

Users issued grants for a specific gateway have the privileges to perform the associated actions on that gateway. See Issuing Gateway Grants to issue gateway grants.

Grant Name	Description	Can be Issued To	Associated Actions
Manage Gateway	People issued this grant are allowed to manage API deployments to this gateway and manage the gateway itself.	Gateway Managers	GatewayManage GatewayViewAllDetails GatewayDeploy GatewayRequestDeploy GatewayApproveDeployRequest GatewayGrantManageGateway GatewayGrantViewGateway GatewayGrantDeployAPI GatewayGrantRequestDeployAPI
View all details	People issued this grant are allowed to view all information about this gateway	Gateway Managers, API Managers, Plan Managers	GatewayViewAllDetails
Deploy to Gateway	People issued this grant are allowed to deploy or undeploy APIs to this gateway.	Gateway Managers, API Managers	GatewayDeploy GatewayRequestDeploy
Request Deployment to Gateway	People issued this grant are allowed to request API deployments to this gateway. Requests must be approved by a Gateway Manager	API Managers	GatewayRequestDeploy
Node Service Account	Gateway Runtime service accounts are issued this grant to allow them to download configuration and upload statistics.	GatewayRuntime	GatewayRetrieveConfiguration GatewayUploadStatistics

Issuing Gateway Grants

Issue gateway grants to users or groups to determine what actions assignees can perform with that gateway. See Understanding Gateway Grants. Grants are issued per gateway; repeat this task for each gateway you want to issue grants for.

Gateway Managers must be issued the Manage Gateway grant to a gateway to issue grants for it.

1. From the Gateways List page, click the name of the gateway for which you want to manage grants.
2. Click the  (Grants) tab.
3. Click the tab that corresponds to the grant you want to issue to users or groups:
   - **Manage Gateway**: Gateway Manager users issued this grant are allowed to manage API deployments to this gateway and manage the gateway itself.

- **View All Details**: API ,Gateway, an Plan Manager users issued this grant are allowed to view all information about this gateway.
- **Deploy to Gateway**: API and Gateway Manager users issued this grant are allowed to deploy or undeploy APIs to this gateway.
- **Request Deployment to Gateway**: API Manager users issued this grant are allowed to request API deployments to this gateway. Requests must be approved by a Gateway Manager.
- **Node Service Account**: Gateway Runtime service accounts are issued this grant to allow them to download configuration and upload statistics.

4. Click **Add Grantee**.

   The Add Grantee dialog appears.

5. From the Add Grantee dialog, select the user(s) or group(s) to which you want to issue the grant. You can select multiple users and groups.

> **Note:**
>
> You cannot select users or groups that already have this grant; they are greyed out in the Add Grantee dialog.

6. Click **Add**.

The user(s) or group(s) are issued the gateway grant you chose.

Working with Deployed Endpoints

Gateway Managers can use the Management Portal to view deployed APIs' details, deploy, redeploy, or undeploy APIs, and approve or reject deployment requests.

Topics

Viewing API Details

You can view the API Request URL, the policies configured for the API, and the service request URL for APIs on your gateway. Gateway Managers must be issued the View All Details or Deploy API grant for an API to view its details.

To view a deployed API's details:

1. From the Gateways List page, select a gateway.
2. Click the  (Deployments) tab for the gateway.
3. Click the tab corresponding to the API's state on the gateway:
   - **Deployed**: Lists APIs currently deployed to the gateway.

- **Requesting**: Lists APIs requesting deployment to the gateway.
- **Waiting**: Lists APIs pending deployment to the gateway.
- **Rejected**: Lists API deployments rejected by a gateway manager.
- **Failed**: Lists failed API deployments.

4. Click the name of the API for which you want to view details, or click the **Expand** icon.

   The Request and Response flows, and the policies configured in each, appear.

5. (Optional) Click the **View Policy** icon to view details about any policy in the flow.

   The View Policy dialog appears, displaying the configuration of and comments describing this policy. You can't make any changes on this dialog.

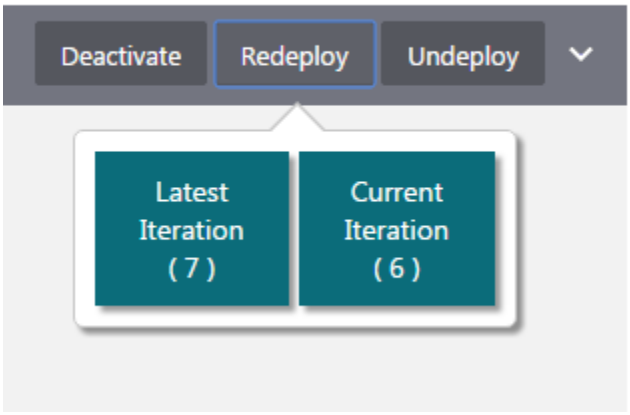
Deploying or Redeploying an API Endpoint to a Gateway

You can deploy or redeploy an API to a gateway to which you have deployment privileges.To deploy or redeploy an API, Gateway Managers be issued the Manage Gateway or Deploy to Gateway grant for the gateway they want to deploy to and the Deploy API grant for the API they want to deploy.

To deploy or redeploy an API:

1. From the Gateways List page, click the gateway you want to deploy or redeploy an API to.
2. Click the  (Deployments) tab.
3. To deploy an API that is not already deployed to the gateway:
   a. Click **Deploy API**.
   b. Use the Filter field to find the API that you want to deploy, and then select the API.
   c. From the Initial Deployment State section, select **Active** to deploy the API in an active state, or select **Inactive** to deploy the API in an inactive state.
   d. (Optional) In the **Description** field, enter comments about the API deployment.
   e. Click **Deploy**.

   The deployment enters a **Waiting** state and the logical gateway definition is updated. The endpoint is deployed the next time gateway node(s) poll the management server for the updated gateway definition.
4. To redeploy an API that is already deployed to a gateway:
   a. Hover over the Production Gateway deployment, and click **Redeploy** when it appears.
   b. Click **Latest Iteration** to deploy the most recently saved iteration of the API, or click **Current Iteration** to redeploy the currently deployed iteration of the API.



c. When prompted, enter comments about why you are redeploying the API, and then click **Yes**.

The deployment enters a **Waiting** state and the logical gateway definition is updated. The endpoint is deployed the next time gateway node(s) poll the management server for the updated gateway definition.

Approving an API Deployment Request

Gateway Managers approve API Deployment requests.

API Managers issued the Request Deployment to Gateway grant can request API deployments to a gateway. These APIs are placed in a **Requesting** state. APIs in this state do not process requests until the deployment is approved.

Gateway Managers must be issued the Manage Gateway grant to approve API deployment requests for a gateway.

To approve an API deployment request:

1. From the Gateways List page, click the gateway you want to approve deployment requests for.
2. Click the (Deployments) tab.
3. Click the **Requesting** tab.
4. Hover over the API you want to approve and click **Approve**.

> **Note:**
>
> You can also click **Reject** to reject the request or **Cancel** to dismiss the request.

The deployment is approved and enters the **Waiting** state. The API is deployed to nodes registered to the gateway when each polls the management service for the latest logical gateway definition.

Undeploying an API

Undeploy an API if you no longer want gateway nodes to process requests for it. Gateway Managers must be issued the Manage Gateway grant or the Deploy API (API) and Deploy to Gateway (gateway) grants to undeploy an API from a gateway.

To undeploy an API:

1. From the Gateways List page, click the gateway you want to undeploy APIs from.
2. Click the  (Deployments) tab.
3. Hover over the API you want to undeploy, and click **Undeploy**.
4. When prompted, enter comments about why you are redeploying the API, and then click **Yes**.

The undeployment request enters the **Waiting** state, which means that the undeployment request is pending. The API is undeployed from nodes registered to the gateway when each polls the management service for the latest logical gateway definition.

Upgrading a Gateway

All gateway nodes are automatically upgraded as a result of the Management Tier upgrade.

Management tier upgrades are user initiated using a script. The script, as part of the upgrade, uploads new artifacts and policies to the database. This trickles down to the gateways on the basis of metadata.

Prior to the upgrade, set the Gateway Node Polling Interval to 120 minutes or more. Once the update completes, reset the polling interval to the previous setting.

Delete a Logical Gateway

Administrators and Gateway Managers can delete logical gateways in the API Platform Cloud Service Management Portal.

You can't delete a logical gateway if nodes are registered or requesting registration to it or if you don't possess the Manage Gateway grant for the gateway. If you can't delete a logical gateway the Delete button is grayed out. Ensure you unregister all nodes from it and that you have the proper grant before trying again.

To delete a logical gateway:

1. On the **Gateways List** page, click the gateway you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes** in the banner to confirm.

4

Manage APIs

In Oracle API Platform Cloud Service - Classic, API Managers manage, secure, and publish APIs with the Management Portal.

This chapter describes the tasks API Managers can perform in the Management Portal. API Managers are people responsible for managing the API lifecycle, which includes designing, implementing, and versioning APIs. API Managers are also responsible for managing API grants and API registrations. API Managers may also be able to deploy or request deployment of their APIs to gateways.

Administrators or API Managers issued the Manage API grant for an API can perform the actions on it described in this chapter.

Topics

Typical Workflow for Managing APIs with Oracle API Platform Cloud Service - Classic

To start managing APIs with Oracle API Platform Cloud Service - Classic, refer to the typical task workflow.

Task	Description	More Information
Create an API	Create an entry for your API in the Management Portal.	Creating an API
Configure the request endpoint	Configure the endpoint to which users and applications send requests to your API.	Configuring the API Request URL
Configure the service request URL	Configure the URL of the API's backend service.	Configuring the Service Request URL
Apply policies	Apply policies to secure, manage traffic, manage interfaces, route, and perform other actions before client requests are passed to your backend services.	Applying Policies
Add overview and documentation text	Describe what your APIs do and provide detailed documentation for your consumers.	Adding Overview Text for an API Documenting an API
Deploy your API to a gateway	Deploy an endpoint for your API to a gateway when its ready to receive requests.	Deploying or Redeploying an API Endpoint to a Gateway
Publish the API to the Developer Portal	Publish API details to the Developer Portal so application developers can discover and subscribe to it.	Publishing an API to the Developer Portal

About the Oracle Apiary Integration

Oracle API Platform Cloud Service - Classic integrates with Oracle Apiary to provide API design and documentation features.

Oracle Apiary provides you with the ability to design APIs using either API Blueprint or Swagger 2.0. From these description files, Oracle Apiary generates interactive documentation and a console for making calls to the APIs from the UI. Oracle Apiary also instantiates a mock service that you can use to interact with the examples provided in the specification file. API Managers can link APIs they have on Oracle Apiary to display interactive documentation, a test console, and mock service details on an API's page in the Developer Portal.

Adding documentation with the Oracle Apiary integration requires a Pro team account. Application Developers viewing Oracle Apiary documentation on the Developer Portal do not need an Oracle Apiary account. Visit http://apiary.io to learn more about Oracle Apiary and its features and to register for an account.

See Adding Oracle Apiary Documentation to an API to add Oracle Apiary documentation to your APIs in the Management Portal.

Understanding the APIs List Page

The APIs List page displays all APIs created in the Management Portal.

Entries for APIs display the following information:

- The name and version of the API.
- The state of publication of the API: **Published**, **Never Published**, or **Unpublished**.
- The state of the API: **Alpha**, **Beta**, **Deprecated**, **Released**, or **Retired**.
- The description of the API.
- The date and time the API was created and which user created it.
- **API Portal URL**: If published, a link to the page on the Developer Portal is displayed. If not published, a Never Published label is displayed.

> **Note:**
>
> The information you see on this page, and the tabs for an API, depends on the grants that you have. For example, if you are an API Manager with the View Details Grant, you will only be able to view the Publication tab.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.
- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even within words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

  > **Note:**
  >
  > If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Creating an API

Create an entry for an API you want to manage in the Oracle API Platform Cloud Service - Classic Management Portal.

There is an option in the Create API dialog box to create a default plan for the API. By default, the name of the plan is the same as the API, but you can edit it as desired in the **Plan Name** field. A version number is also supplied by default, but it is not required. A plan created in this way also automatically has an entitlement for the API, although the entitlement is inactive by default.

> **Note:**
>
> This option only appears if you have the appropriate grants to create plans.

To create an API:

1. From the APIs List page, click **Create API**.
2. In the **API Name** field, enter the name of the API.

   > **Note:**
   >
   > The API name must be at least five characters.

3. In the **Version** field, enter the version of the API. The version numbers are alphanumeric and are limited to 50 characters.
4. (Optional) In the **Description** field, enter a brief description of the API.
5. (Optional) Click the **Create a default plan for this API** option. Accept the default name for the plan or edit it as desired.
6. Click **Create**.

The API is displayed on the APIs page.
See the following topics to complete the API management lifecycle:

- Implementing APIs to implement your API
- Managing API Grants to manage grants to your API
- Publishing APIs to add documentation references and publish your API to the Developer Portal
- Deploying Endpoints to deploy your API to a gateway

Viewing API Details

You can view the details of an API in a side panel available from any of the tabs.

The side panel displays the following details:

- The name and version of the API.

- The description of the API.
- The state of publication of the API, **Published**, **Never Published**, or **Unpublished**.

> **Note:**
>
> This state refers to whether the API has been published to the Developer Portal from the **Publication** tab

- The state of the API: **Alpha**, **Beta**, **Deprecated**, **Released**, or **Retired**.
- The iteration number, which is equal to the number of times changes have been saved for the API.
- The most recent date and time that changes were saved for the API, and name of the user who saved them.

API Managers must be issued the View All Details or Manage API grant for an API to view its details.

To view API details:

1. On the APIs List page, select the API for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the API.

Editing an API Description

You can edit the description of an API in the Management Portal.

To edit the name or description of an API:

1. On the APIs List page, click the API you want to edit.
2. Click the drawer icon to display the side panel.

3. Click the description to edit it.
4. Click **Save**.

Uploading an API Icon

You can upload an icon to visually represent an API in the Management Portal. The icon you upload also represents the API on the Developer Portal if the API is published.

For best results, the image you upload should be 60 pixels by 60 pixels. images with other dimensions may be distorted in the Management and Developer Portals.

PNG and JPEG (`.jpg` and `.jpeg`) image formats are supported.

To upload an icon for an API:

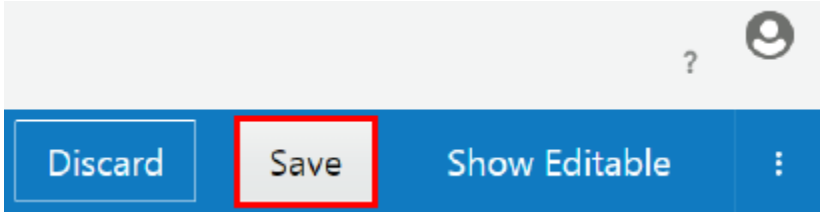
1. On the APIs List page, select an API.
2. Click the drawer icon to display the side panel.

3. Click the icon to the left of the API name in the side panel.

   The icon dialog appears. The **Custom** tab is selected by default.
4. Choose one of the following options:
   a. To upload an API icon, click **Choose File**. Select the image you want to use as the API icon, and then click **Open**. Click **OK** to close the dialog.

> **Tip:**
>
> You can also drag and drop an image onto the **Drop File Here** area on the **Custom** tab.

   b. To revert to the default icon, click the **Default** tab, and then click **OK** to close the dialog.
5. Click **Save**.



The API icon is updated. If you want this icon to represent the API in the Developer Portal, (re)publish the API. See Publishing an API to the Developer Portal.

Cloning an API

Cloning an API allows a management portal user to easily make a copy of an existing API.

You cannot clone an API if you do not have privileges to create new APIs. For example, if you only have a view grant, you cannot clone an API. You also cannot clone an existing API if there are any pending updates to the API which have not been saved. When the updates are saved or discarded, you can then clone the API.

There are options for two attributes of the original API that you can choose to clone:

- **API Implementation**: If this option is selected, all the policy configuration from the original API is copied to the cloned API. However, if the original API contains sensitive security information, such as passwords, in the policies, this sensitive information is not available to the browser and a warning message appears. All the policies are still copied, but you must update the cloned API with new password entries after the clone completes. If this option is not selected, the API implementation is initialized with defaults equivalent to creating a new API.
- **Publication — Configuration**: If this option is selected, the Overview and Documentation configuration is copied to the cloned API, including any file-based artifacts from the original API. If the original API is published, this state is not copied to the new API. The new API is always in the unpublished state and without a portal vanity name. You must update the cloned API with a unique portal vanity name and publish it.

The API State is not copied from the original API. The default is Alpha, which is the default for newly created APIs. You can choose to assign a different state if you wish. Also note that other information associated with an API that is not listed above is not copied to the cloned API. Deployments, grants, and registrations are set to the defaults for a newly created API, and can be updated after the clone is complete.

To clone an API:

1. On the APIs List page, select an API.
2. Click the drawer icon to display the side panel.

3. Click the **Clone** button.

   The Clone API dialog opens.
4. Enter a version number for the cloned API.
5. Deselect the **API Implementation** and **Publication — Configuration** options as appropriate.
6. Click the **API State** list and select the desired state for the cloned API.
7. Click the **Clone** button.

Changing the State of an API

Use this status to inform people if an API is in a Alpha, Beta, Deprecated, Released, or Retired state. You can also include comments about this state that people can view on the Developer Portal.

To change the state of an API:

1. From the APIs List page, select the API you want to update.
2. Click the drawer icon to display the side panel.

3. From the list, select the state you want to assign to the API. The available options are:
   - **Alpha**: API published for preliminary testing. It might not be feature complete and has bugs.
   - **Beta**: API is published for beta testing. It is mostly feature complete but has bugs.
   - **Deprecated**: API is deprecated in favor of a newer version or another API.
   - **Released**: API is released and ready for production use.
   - **Retired**: API is retired. Typically an API enters this state after the depreciation period ends or if it is no longer supported.

   Alpha is the default option.
4. **(Optional)** Describe why the API is in this state.
5. Click **Yes**.

The API's state is changed. This is reflected in both the Management and Developer portals.

Linking an Oracle Apiary Specification

You can link an Oracle Apiary specification to an API. Apiary specifications can contain documentation and a console for making calls to the API.

When you connect to Oracle Apiary, it checks whether documentation exists for the API. If it does not, the **Use As API Documentation** option in the dialog is selected automatically so that the Apiary specification is used as documentation for the API. If documentation already exists for the API, the option is not selected. If you then select it manually, you are warned that the existing documentation will be deleted.

Topics

- Linking an Oracle Apiary Specification to an API

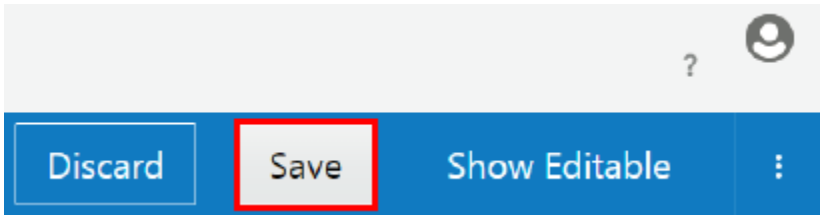
Linking an Oracle Apiary Specification to an API

Use this procedure to connect an Oracle Apiary Specification to an API.

To add an Oracle Apiary Specification to an API:

1. On the APIs List page, select an API.
2. Click the **Apiary** button.

   The Apiary Documentation dialog appears, allowing you to browse Specifications on Oracle Apiary.
3. Select an API Project and then click **Connect**.
4. Click **Save**.



If you have previously published the API, you must republish it to see the Apiary specification.

Implementing APIs

After you create an API, apply policies to configure the request and response flows.

Policies in the request flow secure, throttle, route, manipulate, or log requests before they reach the backend service; polices in the response flow manipulate and log responses before they reach the requesting client. See Configuring the API Request URL and Configuring the Service Request URL to configure mandatory policies. These are added for you when you create an API in the UI.

You must have the Manage API grant for an API to apply or configure policies for it.

Topics

- Understanding Policies
- Configuring the Request Pipeline
- Configuring the Response Pipeline
- Policy Placement
- Applying Policies
- Working with Draft Policies
- About Using Groovy in Policies

Understanding Policies

You apply any number of policies to an API definition to secure, throttle, route, or log requests sent to your API. Depending on the policies applied, requests can be rejected if they do not meet criteria you specify when configuring each policy.

Policies are executed in the order they appear on the Request and Response tabs. A policy can be placed only in certain locations in the execution flow. Most policies can be placed in the request flow; only the redaction, logging, and groovy script policies can be placed in the response flow. See Policy Placement to learn more about policy placement.

Oracle API Platform Cloud Service - Classic provides these types of policies:

- **Security**: policies that determine who can send requests to your services. See these topics to learn more:
  - Applying OAuth 2.0 Policies
  - Applying Key Validation Policies
  - Applying Basic Authentication Policies
  - Applying IP Filter Validation Policies
  - Applying CORS Policies
- **Traffic Management**: policies that manage the volume of traffic sent to your services. See these topics to learn more:
  - Applying API Throttling–Delay Policies
  - Applying Application Rate Limiting Policies
  - Applying API Rate Limiting Policies
- **Interface Management**: policies that manage the service interfaces clients are permitted to access. See these topics to learn more:
  - Applying Header Field Filtering Policies
  - Applying Interface Filtering Policies
  - Applying Redaction Policies
  - Applying Header Validation Policies
  - Applying Method Mapping Policies
  - Applying REST to SOAP Policies
- **Routing**: policies that route requests to different service URLs depending on the requesting application, the resource requested, and other conditions. See these topics to learn more:
  - Applying Header-Based Routing Policies
  - Applying Gateway-Based Routing Policies
  - Applying Application-Based Routing Policies
  - Applying Resource-Based Routing Policies
- **Other**: policies not belonging to one of the above categories. See these topics to learn more:

- Applying Service Callout 2.0 Policies
- Applying Logging Policies
- Applying Groovy Script Policies

Configuring the Request Pipeline

You can add policies in the request flow between the request from a client and when the request is sent to the backend service.

When viewing an API in the Management Portal, click the **Request** tab to view a top-down visual representation of the request flow.

The API Request URL is the endpoint clients send requests to. This represents when an endpoint deployed to a gateway receives a request. You apply any number of policies between the gateway receiving a request. See Policy Placement for a list of policies that can be placed in the response pipeline and their valid locations. Policies execute in order, with the uppermost policy first, followed by the next policy, and so on, until the request is rejected or sent to the backend service if all policy conditions are met. The Service Request URL is where requests meeting the policy criteria are sent to your service.

Configuring the Response Pipeline

You can add policies in the response flow between the response from a backend service and when the response is sent to the client.

When viewing an API in the Management Portal, click the **Response** tab to view a top-down visual representation of the response flow.

> **Note:**
>
> The Service Response and API Response entries can't be edited. The Service Response and API Response entries are visual representations of the response sent from the backend service to the gateway and the response sent to the requesting client, respectively.

The service response happens first. The response from the backend service is always the first entry in the outbound flow. You can place additional policies in this flow. Policies execute in order, with the uppermost policy first, followed by the next policy, and so on, until the response is sent back to the client. See Policy Placement for a list of policies that can be placed in the response pipeline and their valid locations.

Policy Placement

You can place policies only in specific positions in your API implementations.

The following table describes where and in what order polices can be placed in the request and response flows for your APIs. Because policies are executed sequentially, the order in which they appear is important. Each policy is assigned a number, where 1 the first policy and 100 is the last policy in the flow. The API Request is always first in the request flow (with a value of 5) and the Service Request is always last (with a

value of 100). Valid placement for polices within this range is determined by its placement value. For example, key validation policies (with a placement value of 10) can only be placed after API Requests but before all other polices (their values greater than 10).

Policies with the same value can be placed in any order relative to their position in the flow. For example, interface filtering and header validation polices can be placed in any order as long as they are placed after a key validation policy but before a header-based routing policy. Policies with multiple values, such as the Groovy Script policy, can be placed in any of the listed positions.

Policy Type	Valid Request Flow Placement	Valid Response Flow Placement
API Request	5	-
Service Response	-	150
Key Validation	10	-
Basic Auth	11	-
Oauth 2	11	-
API Rate Limiting	30	-
API Throttling - Delay	30	-
Application Rate Limiting	30	-
Interface Filtering	30	-
Service Callout	30	-
Method Mapping	30	-
CORS	30	-
Redaction	30	150
Header Validation	30	-
IP Filtering	30	-
Service Level Authorization	40	-
Header Based Routing	50	-
Resource Based Routing	50	-
Application Based Routing	50	-
Gateway Based Routing	50	-
Groovy Script	30,40,50	150
Logging	30,40,50	150
API Response	-	150
Service Request	100	-

Applying Policies

Apply policies to an API to secure, throttle, route, or log requests sent to it. Depending on the policies applied, requests can be rejected if they do not meet criteria you specify when configuring each policy.

You must have the Manage API grant for an API to apply or configure policies for it.

Topics

Configuring the API Request URL

The API Request URL is the endpoint to which users or applications send requests for your API. You configure part of this URL.

The full address to which requests are sent consists of the protocol used, the gateway hostname, the API Request endpoint, and any private resource paths available for your service. An example API Request URL is `http://example.com:8001/Energy1/estimate/4859634`, where:

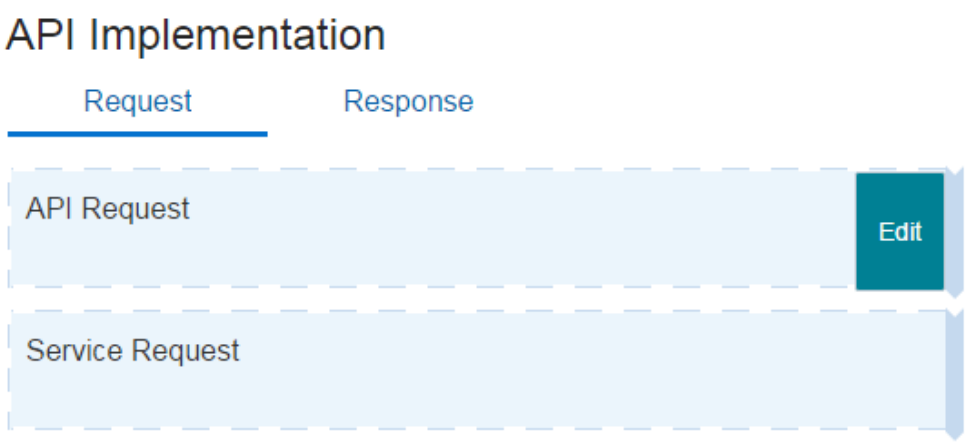
http is the protocol over which the gateway receives requests.
http://example.com:8001/ is the hostname and port of the gateway node instance to which this API is deployed.
Energy1 is the API endpoint you configure. Anything beyond the API endpoint is passed to the backend service.

/estimate/4859634 is the private resource path of the API. This is the resource requested of the backend service.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure the API Request URL:

1. Hover over **API Request** and then click **Edit**.



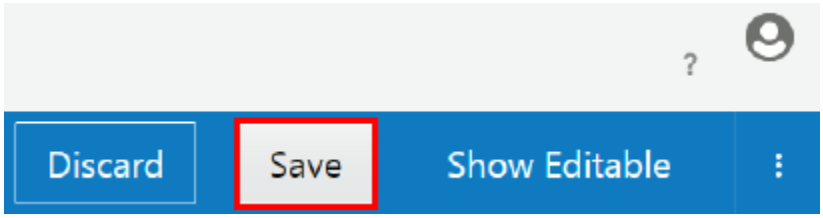
   The Edit Policy dialog appears.

2. Provide the following information to configure the request:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Protocol** list, select the protocol over which the gateway receives requests for the API. Your options are **HTTP**, **HTTPS**, or **HTTP & HTTPS**.
   d. In the **API Endpoint URL** field, enter the endpoint URL for this API. This is case sensitive; `energy` and `Energy` are considered different endpoints.

      NOT_SUPPORTED:

      The following endpoints are reserved for internal use: **/apiplatform** and **/prm_pm_rest**. Do not use these values as endpoints for your APIs.

   e. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.
3. Click **Save**.



The API Request URL is configured.

> **Important:**
>
> The API Implementation is not valid until you also configure the service request, as described in Configuring the Service Request URL. You cannot deploy it to a gateway until you configure the service request.

The API Request URL is not displayed in the Developer Portal when the API is published. You must provide it in the documentation so Application Developers know where to send requests. See Documenting an API.

Configuring the Service Request URL

The service request is the URL at which your backend service receives requests.

When a request meets all policy conditions, the gateway routes the request to this URL and calls your service. Note that the service request URL can point to any of your service's resources, not just its base URL. This way you can restrict users to access only a subset of your API's resources.

You can also control which requests are passed through by configuring the headers. By default, all headers are passed through except Proxy-Authorization, Authorization, Content-Length, Transfer-Encoding, Host, OSCGOAuthBearer. OSCGOAuthMAC, Anonymous, OSCGProxy-Authorization, OSCGSoapHeader, and OSCGAppKeyHeader.

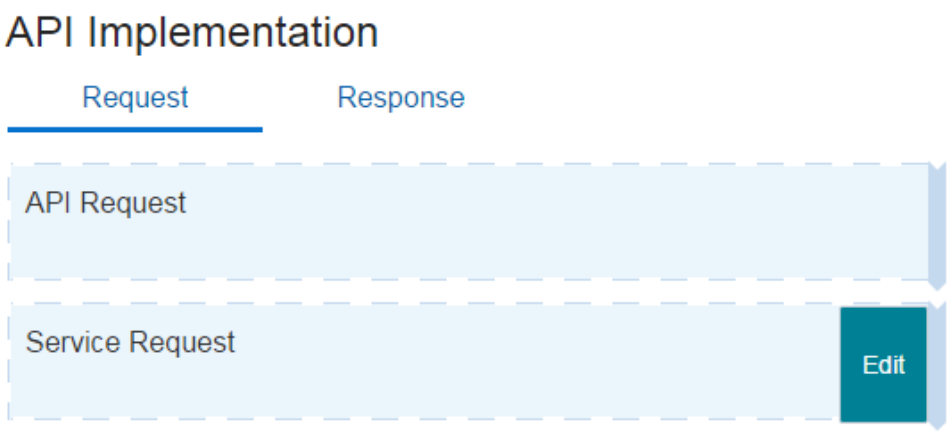
REST and SOAP backend services are supported.

> **Note:**
>
> Some policies, such as OAuth 2.0, method mapping, and interface filtering, were designed to work primarily with REST services; it may not make sense to apply these policies to SOAP services. See Are SOAP APIs Supported?.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure the service request:

1. Hover over the **Service Request** region, and then click **Edit**.



The Edit Policy dialog appears.

2. From the Edit Policy dialog:
   a. Expand the **Configure Headers** section.
   b. In the **all the Headers** list, select **Pass through** or **Drop**.
   c. If you selected **Pass through**, click the **Drop Headers** field to list header names to be deleted. Click in the field and select a header to drop. Repeat for each header you want to drop. You can also type the header name in the field and press **Enter**.
   d. If you selected **Drop**, click the **Pass Through Headers** field to list header names to be passed through. Click in the field and select a header to be passed through. Repeat for each header you want to drop. You can also type the header name in the field.
   e. To add or update headers, click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field. .
   f. **(Optional)** Click the **+** (Add new Header) icon to add more headers as needed. Repeat Step 2h for each header you add.
   g. In the **Service** section, choose one of the following:
      - **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.
      - **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

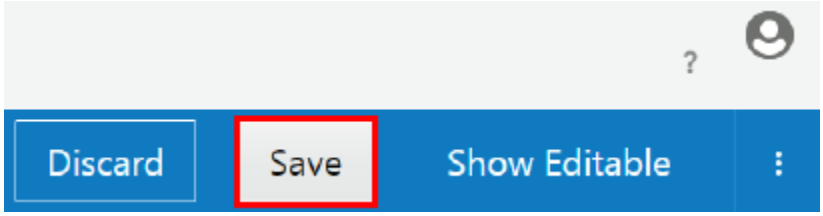
        Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

h. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

i. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The service request URL is configured.

> **Important:**
>
> The API Implementation is not valid until you also configure the API request, as described in Configuring the API Request URL. You cannot deploy it to a gateway until you configure the API request.

Applying OAuth 2.0 Policies

Use an OAuth 2.0 policy to secure an API using OAuth 2.0.

A client application is authenticated by the identity provider and receives an access token. The client application sends the token with requests to the gateway, which acts as an OAuth enforcer and validates the token. See Configure OAuth Providers. If the token is valid, the request is passed on to the protected resource. If the token isn't valid, the request is rejected.

In addition to validating tokens, you can limit access to your APIs by scope. You can also limit access per HTTP method (GET, PUT, POST, and DELETE) to specific scopes. For instance, you can allow only tokens issued for .WRITE scopes for POST operations.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an OAuth 2.0 policy:

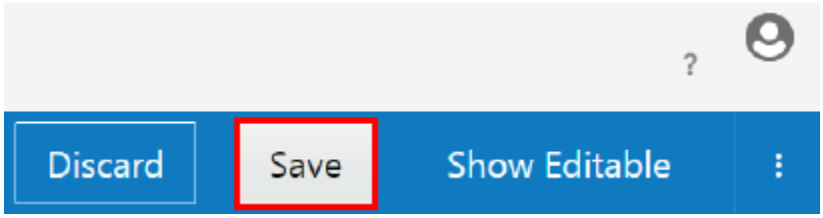
1. In the Available Policies region, expand **Security**, hover over **OAuth 2.0**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.
   d. Click **Next**.
   e. From the **Scope Enforcement Options** section, select **Any** to allow all scopes (passing all requests with a valid token), select **At Least One** to pass requests containing at least one of the scopes you specify, or select **HTTP Method Restricted** to define which scopes are allowed to access resources based on HTTP method: GET, PUT, POST, and DELETE.
   f. If you selected **At Least One**, enter a scope, like `.READ`, into the **Valid Scope** field, and then press the **Enter** key.

      You can enter multiple scopes. Requests (with valid tokens) from any scopes you enter in a single field are passed to the backend service. All other requests are rejected, even if the tokens are valid. Press **Enter** after entering each scope. Remove a scope by clicking the **X** next to it.
   g. If you selected **HTTP Method Restricted**, select an HTTP method from the **Method** list, enter a scope, like `.READ`, into the **Valid Scope** field, and then press the **Enter** key.

      You can enter multiple scopes. The gateway validates that the scope claim (named “scope”) is present in the access token. Requests (with valid tokens) from any scopes you enter in a single field for an HTTP method are passed to the backend service. All other requests are rejected, even if the tokens are valid. Press **Enter** after entering each scope. Remove a scope by clicking the **X** next to it.
   h. **(Optional)** Click the **+** (Add Scope) icon to add an additional scope enforcement condition. Repeat Step 2f if you selected **At Least One** or Step 2g if you selected **HTTP Method Restricted** to configure additional scope enforcement options.
   i. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.
3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Key Validation Policies

Use a key validation policy when you want to reject requests from unregistered (anonymous) applications.

Keys are distributed to clients when they register to use an API on the Developer Portal. At runtime, if they key is not present in the given header or query parameter, or if the application is not registered, the request is rejected; the client receives a `400 Bad Request` error if no key validation header or query parameter is passed or a `403 Forbidden` error if an invalid key is passed.

This policy can be added only to the request flow.

You can only apply the key validation policy once per API.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

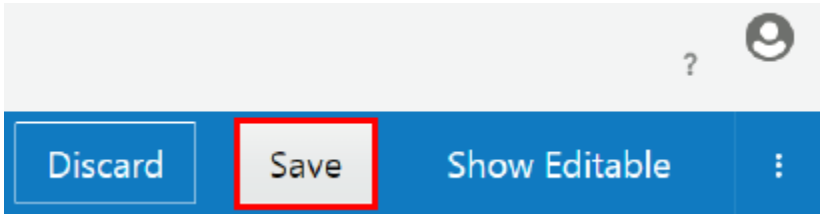
To configure a key validation policy:

1. In the Available Policies region, expand **Security**, hover over **Key Validation**, and then click **Apply**.

   > **Note:**
   >
   > If you have already applied a key validation policy to this API, the Apply button is disabled. You can only apply the key validation policy once per API.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.
   d. Click **Next**.
   e. From the **Key Delivery Approach** region, select **Query Parameter** if the client is to pass the key as a query parameter, or select **Header** if the client is to pass the key in a header.
   f. In the **Key Query Parameter** or **Key Header** field, enter the name of the query parameter or header with which clients must pass the key. The request is rejected if the parameter/header is not present, if the key is not present, or if the key is invalid.
   g. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.
3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying IP Filter Validation Policies

Use the IP Filter Validation policy to control which IP Addresses can successfully send requests to your API. IPv4 and IPv6 addresses are supported.

The IP address originated by the client is received from the `HTTPRequest`. This policy checks if the address matches allowed or disallowed IP addressed configured as part of the policy. Upon finding a match, it takes appropriate action as configured. Nonstandard HTTP headers such as `X-ProxyUser-Ip`, `X-Forwarded-For` , or `HTTP_X_CLUSTER_CLIENT_IP` are not supported.

This policy can be added only to the request flow, but it cannot be placed first in the flow. Other security polices must be placed before it. If you want to filter out requests from potentially malicious sources before they can be rejected by the IP Filter Validation policy, use the load balancer provisioned with your service instance. See Signing in to the Load Balancer Console for Your Instance.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an IP filter validation policy:

1. In the Available Policies region, expand **Security**, hover over **IP Filter Validation**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. From the **IP Address Conditions** list, select one of the following:

      - Select **PASS** to pass the request if any of the IP Address conditions are met.
      - Select **REJECT** to reject the request to the API if any of the IP Address conditions are met.

> **Tip:**
>
> The requester receives a 403 Forbidden HTTP status code and an “IP filter validation failed” message in response to a rejected request.

f. In the **IP Address** field, select either **IPv4** or **IPv6**.

g. In the **Expression** field, select the expression used to evaluate the addresses entered into the **Value** field. The following expressions are available:

- **=** (is equal to)
- **!=** (is not equal to)
- **Inside** (is inside of specified range), like 12.12.1.100-12.12.12.122
- **Regex Match**(is equal to an address included in the supplied regex mask. The Regex is made up of wildcard characters. If the IP of the host that is making the request to the API matches the wildcard, then it is passed or rejected. If no condition is evaluated to true, the opposite of the action gets executed.), like 12.12.1.*
- **CIDR Notation** (is equal to an address included in the supplied CIDR notation expression), like 12.12.12.123/24, which includes 12.12.12.0-12.12.12.255

h. In the **Value** field, enter the IP Address value you want the expression to evaluate. The values you enter in this field differ based on the expression you chose in the previous step.

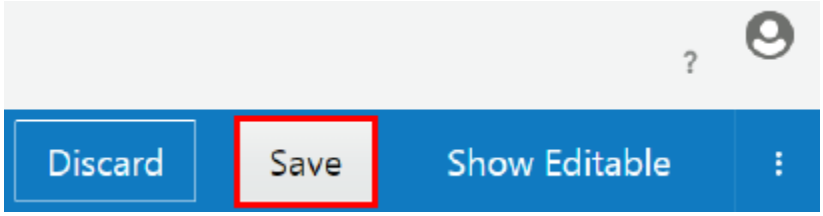
For example, if you chose **Inside**, enter an IP address to begin and to end the range. If the IP address from which this request is sent is within this range, the request is either passed or rejected based on the behavior you specified.

If the IP address is not within this range, each subsequent IP address condition is evaluated. When an IP address condition evaluates as true, the request is either passed or rejected based on the behavior you specified. If no condition is evaluated to true, the opposite of the action gets executed.

i. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2e through Step 2g to populate the data for any additional conditions you add.

j. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Basic Authentication Policies

Use a basic authentication policy to secure an API using the basic authentication protocol. The policy validates the value of the `Authorization` HTTP header against the identity store that the gateway node is configured against.

After successful authentication, the gateway executes the next policy in the flow or sends the request to the backend service. If authentication fails, the client receives a `401 Unauthorized` error.

This policy is not compatible with any other authentication security policy (like OAuth 2.0); only one of these can be present in each API's request flow.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To apply a basic authentication policy:

1. In the Available Policies region, expand **Security**, hover over **Basic Auth**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

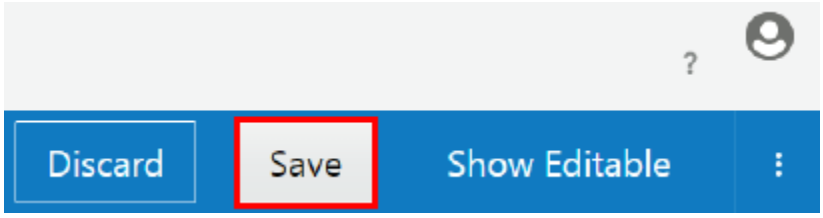
   e. From the Authenticated Users section, select **Specific Users** to pass requests from specific authenticated users or groups, or select **All Users** to pass requests from all authenticated users.

   f. If you selected **Specific Users**, enter each user or you want to be able to call this API into the **Account** field, and then press the **Enter** key. Requests from all other users or groups are rejected.

      Click the **+** icon to add another row to enter user accounts into.

   g. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying CORS Policies

Use a CORS policy to specify which domains are allowed to send requests to your service.

By default, the gateway rejects requests from domains other than its own. Many use cases require that services are invoked from other domains. Requests can be sent with an `Origin` header that includes the requesting domain. When applied, the CORS policy reads the value in the `Origin` header and compares it to allowed domains you specify. If these match, the request is passed to the next policy or the backend service and this header is sent with the response:

```
Access-Control-Allow-Origin: *
```

You can configure the policy to allow requests from specific domains or from all domains.

> **Note:**
>
> Other CORS features are not supported in this release.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a CORS policy:

1. In the Available Policies region, expand **Security**, hover over **CORS**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.
   d. Click **Next**.
   e. To specify domains from which requests are allowed, ensure that **Specific Domains** is selected. Type the domains that you want to allow, and press **Enter** after each domain.

> **Tip:**
>
> Click the **Add** (**+**) icon to add a field. This way you can group similar domains in their own field.

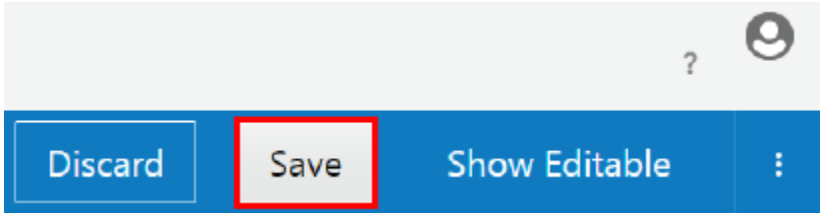
Requests from domains that you haven't explicitly allowed receive this `403 Forbidden` message in response:

```
Origin not allowed.
```

f. To allow requests from all domains, click **All Domains**.

g. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying API Throttling–Delay Policies

Use an API throttling-delay policy to control the global volume of requests to an API by delaying requests exceeding a threshold you set.

API Throttling prevents the abuse of an API. For example, if you want an API to accept only 50 requests per second, all requests received after the 50th during that second are delayed by the time you specify. Throttling is applied to all API requests, regardless of the source.

The limits that you set can be applied to the entire gateway or to each node within the gateway. For example, if you have a gateway with two nodes, and you set a limit for each API of 100 requests per minute, you can choose to apply the limit to each node in the gateway so that each node has 100 requests per minute. You can also apply the limit to the gateway, so that each node has 50 requests per minute.

This policy is different than the API rate limiting policy; requests exceeding the threshold set for that policy are rejected instead of delayed. See Applying API Rate Limiting Policies.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an API throttling-delay policy:

1. In the Available Policies region, expand **Traffic Management**, hover over **API Throttling–Delay**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. In the **Allow API Throttling** field, select **per Logical Gateway** or **per Node**.

   f. In the Condition 1 section, choose one of the following expressions to use to evaluate the condition: **Greater than**, **Greater than or equal to**, or **Is between**.

   g. Enter a positive integer.

   h. From the list, select the time period during which this condition applies. The following time periods are available:

      - **Per second**
      - **Per minute**
      - **Per hour**
      - **Per day**
      - **Per week**
      - **Per month**

      Requests meeting the conditions you specify are delayed. For example, if you specified `Greater than`, `50`, and `Per second`, requests 1 to 50 in a given second are passed as usual. Requests 51+ are delayed by the time period you specify in the following steps.

   i. In the **Delay requests by** field, enter a value to delay requests by. This value must be an integer.

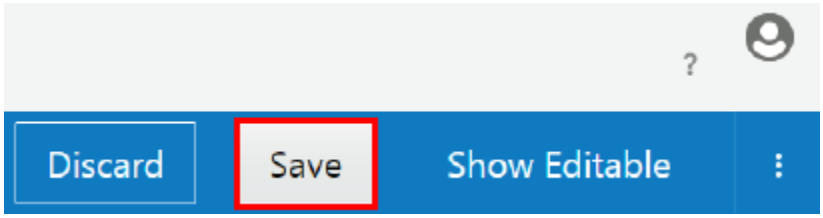
   j. From the list, select **Milliseconds** or **Seconds**.

      Requests meeting the policy conditions are delayed by this time period. For example, if you specified `5 Seconds`, throttled requests are passed five seconds after each is received.

   k. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2e through Step 2i for each condition you add.

   l. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Application Rate Limiting Policies

Use an application rate limiting policy to limit the amount of requests an API allows from each application over time periods that you specify. This time period is defined in seconds, minutes, hours, days, weeks, or months.

The limits that you set can be applied to the entire gateway or to each node within the gateway. For example, if you have a gateway with two nodes, and you set a limit for each application of 100 requests per minute, you can choose to apply the limit to each node in the gateway so that each node has 100 requests per minute. You can also apply the limit to the gateway, so that each node has 50 requests per minute. Gateways reject requests from a given application exceeding any of the thresholds you set.

This policy is different than the API rate limit policy. This policy counts requests from each application separately toward each policy condition. The API rate limit policy counts all requests to an API, regardless of the requesting application, toward each policy condition. See Applying API Rate Limiting Policies.

> **Note:**
>
> This policy must be paired with a key validation policy. The gateway determines which application is sending a request by reading the app key sent with the request. Application rate limit policies have no effect if the API does not also have a key validation policy applied.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an application rate limiting policy:

1. In the Available Policies region, expand **Traffic Management**, hover over **Application Rate Limiting**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

d. Click **Next**.

e. In the **Allow Application Rate Limiting** field, select **per Logical Gateway** or **per Node**.

f. In the **Rate Limit Per Application** field, enter a positive integer.

g. From the **Time Interval** field, select the time period during which this condition applies. The following time periods are available:

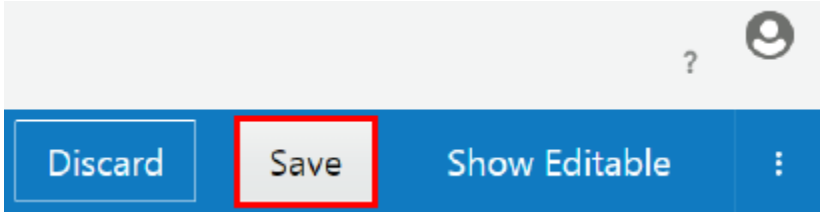
- **Second**
- **Minute**
- **Hour**
- **Day**
- **Week**
- **Month**

For example, if you enter 100 and select **Minute**, the first one hundred requests received during a single minute from one application continue to the next policy or are routed to the backend service, depending on the API's implementation. Any subsequent requests received from the same application during the same minute are rejected. Requests received from other applications are passed if the limit has not been reached for that application.

h. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2d and Step 2e for each condition you add. Requests are rejected if at least one of the thresholds you set are exceeded. Requests are passed if none of the thresholds you set are exceeded.

i. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying API Rate Limiting Policies

Use an API rate limiting policy to limit the total number of requests an API allows over a time period that you specify. This time period is defined in seconds, minutes, hours, days, weeks, or months.

The limits that you set can be applied to the entire gateway or to each node within the gateway. For example, if you have a gateway with two nodes, and you set a limit for each API of 100 requests per minute, you can choose to apply the limit to each node

in the gateway so that each node has 100 requests per minute. You can also apply the limit to the gateway, so that each node has 50 requests per minute. Gateways reject requests exceeding any of the thresholds you set.

This policy is different than the application rate limit policy. This policy counts all requests to an API, regardless of the requesting application, toward each policy condition. The application rate limit policy counts requests from each application separately toward each policy condition. See Applying Application Rate Limiting Policies.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an API rate limiting policy:

1. In the Available Policies region, expand **Traffic Management**, hover over **API Rate Limiting**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. In the **Allow API Rate Limiting** field, select **per Logical Gateway** or **per Node**.

   f. In the **API Rate Limit** field, enter a positive integer.

   g. From the **Time Interval** list, select the time period during which this condition applies. The following time periods are available:

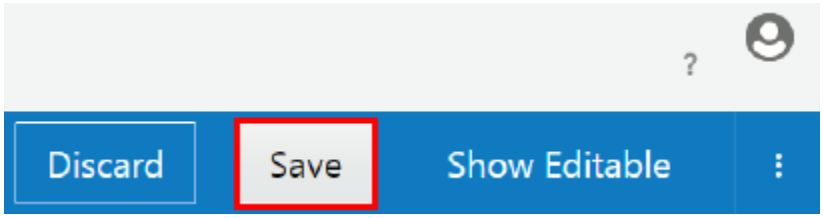
      - **Second**
      - **Minute**
      - **Hour**
      - **Day**
      - **Week**
      - **Month**

      For example, if you enter `100` and select `Minute`, the first one hundred requests received during a single minute continue to the next policy or are routed to the backend service, depending on the API's configuration. Any subsequent requests received during the same minute are rejected.

   h. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2e and Step 2f for each condition you add. Requests are rejected if at least one of the thresholds you set are exceeded. Requests are passed if none of the thresholds you set are exceeded.

i. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Header Field Filtering Policies

Use the header field filtering policy to filter the request headers for length and format. It can be used for security or to reduce the occurrences of failures or errors at the service layer. For example, an API Manager would use this to stop excessively large headers in requests to prevent inadvertent or malicious attacks to a service. Requests are rejected when selected header values are too large. Once a condition is violated, processing stops. Duplicate conditions are not allowed.

The value in the Value is longer than field must be an integer. Only bytes and kilobytes are units of size.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

This policy can be added only to the request flow.

To configure a header field filtering policy:

1. In the Available Policies region, expand **Interface Management**, hover over **Header Field Filtering**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. In the **Resources** field, enter the resources you want to use as conditions, and then press `Enter`. You can enter as many resources as you like.

> **Note:**
>
> Wildcards in resource paths are supported. For example, when entering `/animals/*` as the resource condition, requests to `/animals/cats` and `/animals/dogs` meet the specified resource condition.

e. In the **Methods** field, select the methods to filter by. Select **ANY** to filter by resource only.

f. In the **Header Names** field, enter the desired headers. Press `Enter` after each header name.

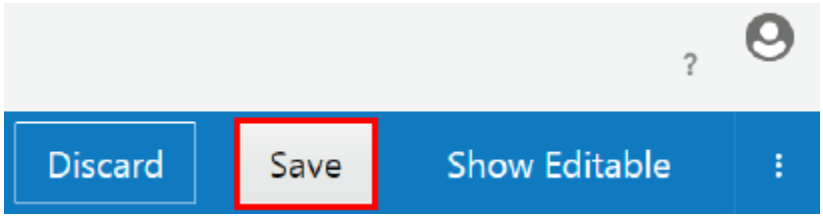
g. In the **Value is longer than** field, enter an integer. In the units list, select **Bytes** or **Kilobytes**.

h. (Optional) Click the **+** (Add a new header condition) icon to add additional header conditions. Repeat step 2f and step 2g for each header condition you add.

i. (Optional) Click the **+** (Add a new condition) icon to add additional conditions. Repeat step 2d through step 2g for each resource and method combination you add.

j. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Interface Filtering Policies

Use an interface filtering policy to filter requests based on the resources and methods specified in the request.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an interface filtering policy:

1. In the Available Policies region, expand **Interface Management**, hover over **Interface Filtering**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

d. Click **Next**.

e. From the list, select **Pass** to pass, or select **Reject** to reject, requests containing the resource and method combinations specified in the policy. **Pass** is the default option.

f. In the **Resources** field, enter the resources you want to use as conditions, and then press **Enter** You can enter as many resources as you like.

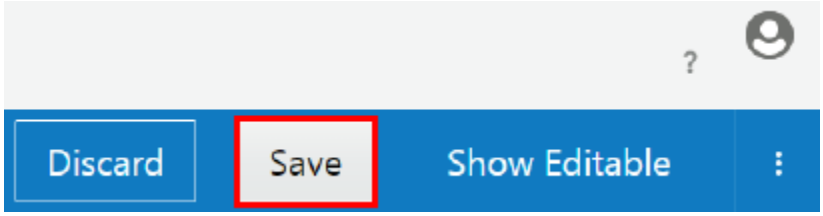
> **Note:**
>
> Wildcards in resource paths are supported. For example, when entering `/animals/*` as the resource condition, requests to `/animals/cats` and `/animals/dogs` meet the specified resource condition.

g. In the **Methods** field, select the methods to filter by. Select **ALL** to filter by resource only.

h. (Optional) Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2f and Step 2g for each resource and method combination you add.

i. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Redaction Policies

Use a redaction policy to limit or remove certain fields and headers that appear in the request or response payload. In the request flow, you can control the header, queries, and payload contents before the backend service is invoked. In the response flow, you can control the response headers, queries, and payload content sent to the client. This policy can be added to the request or response flows.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a field redaction policy:

1. In the Available Policies region, expand **Interface Management**, hover over **Redaction**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request or response flow.

   d. Click **Next**.

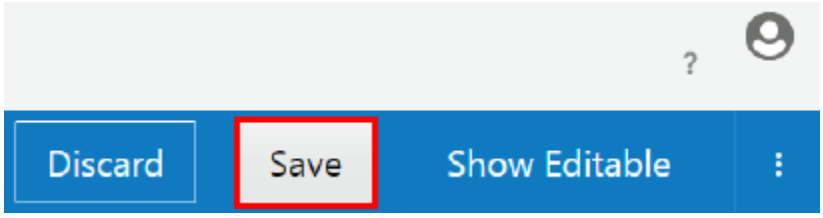
   e. From the **Redaction Conditions** list, select **Include Only** to include only the specified headers or fields in the response, or select **Exclude** to exclude the specified headers or fields from the response.

   > **Note:**
   >
   > You can either include or exclude the whole fields. You cannot include specific values in the fields or exclude specific values from the fields.

   f. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2e for each condition you add.

   g. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Example 4-1 Examples

For the following examples:

- An `exclude` operation means that the referenced nodes and all of their children get deleted. If there is no match, the payload remains intact.
- An `include only` operation means that only the referenced nodes and their children are kept. If there is no match, the payload is emptied.

JsonObject processing example

Input body

```
{
 "a": {
    "b": [ 1, 2, 3 ],
    "c": {
      "d": true
    },
    "e": [
      {
        "f": {
          "g": "11"
        },
        "h": [
          {
            "i": null,
            "j": 2
          }
        ],
        "g": "22"
      },
      {
        "f": 12,
        "h": ["j", "13"]
      },
      {},
      {
        "h": { "j" : "33"}
      }
    ]
  }
}
```

Rule	Referenced fragments	Result

operation: exclude
condition: a.c
	"c": {
	"d": true
	}
		{
		"a": {
		"b": [1, 2, 3],
		"e": [
		{
		"f": {
		"g": "11"
		},
		"h": [
		{
		"i": null,
		"j": 2
		}
		],
		"g": "22"
		},
		{
		"f": 12,
		"h": ["j", "13"]
		},
		{},
		{
		"h": { "j" : "33"}
		}
		]
		}
		}

operation: include only condition: a.x	references nothing	{ }

JSONArray processing example

Input body

```
[
 {
    "a": 0
 },
 {
    "b": [ 1, 2, 3 ],
    "c": {
      "d": true
     }
 },
 {},
 {
    "e": [
```

```
      {
        "f": {
          "g": "11"
        },
        "h": [
          {
            "i": null,
            "j": 2
          }
        ],
        "g": "22"
      },
      {
        "f": 12,
        "h": ["j", "13"]
      },
      {
        "h": { "j" : "33"}
      }
    ]
  }
]
```

Rule	Referenced fragments	Result

operation: include only
condition: [1].c.d
	`"c": {`
	`"d": true`
	`}`
		`[`
		`{`
		`"c": {`
		`"d": true`
		`}`
		`}`
		`]`

Note:

The array item count will be changed in this ca

Rule	Referenced fragments	Result
 s e .		

XML Processing example

Input body

```
<?xml version="1.0" encoding="UTF-8"?>
   <a>
      <b>
         <element>1</element>
         <element>2</element>
         <element>3</element>
      </b>
      <c>
        <d>true</d>
      </c>
      <e>
         <element>
            <f>
              <g>11</g>
            </f>
            <g>22</g>
            <h>
              <element>
                 <i>null</i>
                 <j>2</j>
              </element>
            </h>
        </element>
        <element/>
        <element>
            <f>12</f>
            <h>
               <element>j</element>
               <element>33</element>
            </h>
        </element>
        <element>
            <h>
               <j>33</j>
            </h>
        </element>
      </e>
   </a>
```

Rule	Referenced fragments	Result
operation: include only condition: root.a.e.element[0].f	<f> <g>11</g> </f>	<?xml version="1.0" encoding="UTF-8"?> <root> <a> <e> <element> <f> <g>11</g> </f> </element> </e> </root>

Applying Header Validation Policies

Use a header validation policy when you want to pass or reject requests based on the presence of or value of headers sent with the request.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a header validation policy:

1. In the Available Policies region, expand **Interface Management**, hover over **Header Validation**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.
   d. Click **Next**.
   e. Select **Pass** to pass requests meeting the policy criteria, or select **Reject** to reject requests meeting the criteria. **Reject** is the default option.

> **Tip:**
>
> The client receives a 400 Bad request HTTP status code and a “Bad request” message in response to a rejected request.
>
> When validations fail, you can view the details at `apics/analytics/logs`.

f. Select **Any** to pass or reject a request if any of the header conditions you specify evaluate as true. Select **All** to pass or reject a request only if all of the header conditions you specify evaluate as true.

g. Enter a header name into the **Name** field.

h. Choose an expression with which to evaluate the header in the **Operator** field. The following expressions are available:

- **=** (is equal to)
- **!=** (is not equal to)
- **>** (is greater than)
- **<** (is less than)
- **>=** (is greater than or equal to)
- **<=** (is less than or equal to)
- **Is Null** (the header has no value)
- **Is Not Null** (the header is present and has a value)

i. Depending on the expression you selected, enter a relevant value into the **Value** field. Wild cards are not supported. Some expressions, like **Is Null** and **Is Not Null**, do not require values.

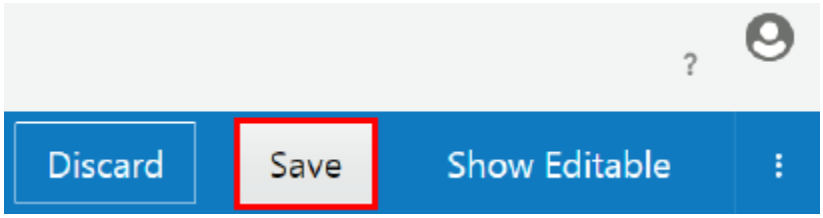
If this header is not present, or has a value other than the one provided, each subsequent header condition is evaluated if you chose Any; if you chose All the request is passed or rejected based on the behavior you selected. If the next header evaluates as true, the request is passed or rejected based on the behavior you selected. If you chose All, the request is passed or rejected as you chose only if all conditions are met.

> **Note:**
>
> If there are duplicate headers, the first condition is validated and then the duplicate condition is validated, even if they are conflicting.

j. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2f through Step 2i to populate the data for any additional conditions you add.

k. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Method Mapping Policies

Use the method mapping policy to change the HTTP method of a request to another method you specify before passing it to the service.

In addition to methods, this policy can also map resources, query parameters, headers, and fields from one value to another that you configure before the request is passed to the service.

This is a transformation policy. It attempts to make changes, but it does not block the request if it cannot perform the transformation. It is meant to be used in conjunction with a validation policy so that unintended requests have already been filtered out by the time this policy is applied.

The Apply Policy dialog for the method mapping policy has two columns: one labeled **API** and another labeled **Service**. When a request is received that meets conditions set in the **API** column, matching fields are replaced with data in the **Service** column when the request is sent to the backend service. For example, if a request is sent with a GET method, but you want to replace it with a POST method in the call to the service, you configure a method mapping policy to change the method from GET to POST.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a method mapping policy:

1. In the Available Policies region, expand **Interface Management**, hover over **Method Mapping**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. In the **Resources(s)** field in the **API** column, select the resources(s) to change.

f. In the **Resource** field in the **Service** column, select the resource to request from the service.

g. In the **Method(s)** field in the **API** column, select the method(s) to change.

h. In the **Method** field in the **Service** column, select the method to send requests with to the service. Select **Keep Same** to retain the value selected in the **API** column.

i. **(Optional)** To map query parameters, expand the **Query Parameters** heading and click the **Add a new query parameter** button (+). In the **API** column, enter the query parameter you want to change. In the **Service** column, enter the query parameter you want to replace its paired API query parameter with and send to the service.

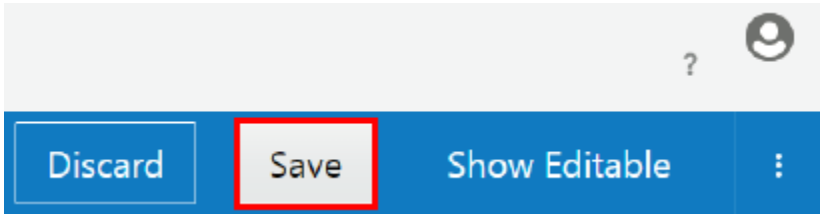
j. **(Optional)** To map headers, expand the **Headers** heading and click the **Add a new header** button (+). In the **API** column, enter the header you want to change. In the Service **column**, enter the header you want to replace its paired API header with and send to the service.

k. **(Optional)** To map fields, expand the **Fields** heading and click the **Add a new field** button (+). In the **API** column, enter the field you want to change. In the **Service** column, enter the field you want to replace its paired API field with and send to the service.

l. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions.

m. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Method Mapping Use Cases

These use cases can serve as examples for using the method mapping policy.

Resource Mapping

If the incoming request resource matches any of the configured API side resources, then it is mapped to the service side resource. You can configure more than one resource on the API side, but it all maps to a single service side resource. For example, if the URL is `http://myserver.com:8001/methodmap/1/business/resource/one/`, the base URL is `http://myserver.com:8001/methodmap/1/` and the resources are `business/resource/one/`. Anything after `business` is considered a resource.

API	Service
/dogs /cats	/animals

In the Resources section, you enter multiple resources in the API column on the left and a single resource in the Service column on the right. In the example table above, if you want dogs and cats to be transformed into animals, enter `/dogs /cats` in the left column and `/animals` on the right.

Method Mapping

Method mapping can be configured to map multiple API methods to a single service method; for example, GET, PUT, and POST should all map to POST in the service. There are other options, such as ANY in the API column on the left and KEEP SAME in the Service column on the right.

API	Service
ANY	POST
POST,PUT	KEEP SAME
ANY	KEEP SAME

In the example table above, if the API side is configured as ANY, then any request method maps to the configured Service method. If the API side is configured as POST,PUT and Service side is configured KEEP SAME, then the mappings apply to POST and PUT controls only.

Mappings

These mappings are applied only if all mapping conditions are met. Otherwise, the next condition is evaluated.

Header Mapping

- **Add Header with Static value** : The incoming request does not have a header, and you want to add a static header such as `xyz=123` to the incoming request header. The API column should be blank, and the Service column should have `xyz=123`.

API	Service
—	`xyz=123`

- **Add Header with Dynamic Value**: The incoming request does not have a header, and you want to add a dynamic header with an `xyz` value to the incoming request header. The API column should be blank, and the Service column should have a statement that takes the value of a specific field from the payload and adds it to the `xyz` header. You can also use queries to set the value of the header. See the table below for examples.

API	Service
—	`xyz=${payload.fields.field1}`
—	`fromQuery=${queries.additional}`

API	Service
—	`fromPayload=${payload.conditions[1].headerValue}`

- **Replace Header Name**: The incoming request has a header, and you want to replace the header with another name and also remove the header from the original request. For example, you want to replace the header `x-header-xyz` with the header `xyz`. The API column should have the header to be replaced, and the Service column should have the replacement value for the header. If the header in the request is not `x-header-xyz`, then the original header is not replaced or removed.

API	Service
`x-header-xyz`	`xyz`
`acno`	`AccountNo`
`apiid=123`	`xyz.application.id=123`
`dept`	`department=${payload.conditions[0].headerName}`

- **Remove Header Name**: The incoming request has a header, and you want to remove the header from the original request. For example, you want to remove the header with the name `xyz` from the request. The API column should have the name of the header, and the Service column should be blank.

API	Service
`xyz`	—
`xyz=123`	—

Query Parameter Mapping

- **Add Query with Static Value**: The incoming request does not have a query, and you want to add a query such as `xyz=123` to the incoming request. The API column should be blank and the Service column should have `xyz=123`.

API	Service
—	`xyz=123`

- **Add Query with Dynamic Value**: The incoming request does not have a query, and you want to add a dynamic query with an `xyz` value to the incoming request header. The API column should be blank, and the Service column should have a statement that takes the value of a specific field from the payload and adds it to the `xyz` query. You can also use headers to set the value of the query. See the table below for examples.

API	Service
—	`xyz=${payload.fields.field1}`
—	`fromHeader=${headers.additional}`

API	Service
—	`fromPayload=${payload.conditions[1].headerValue}`

- **Replace Query Name**: The incoming request has a query, and you want to replace the query parameter with another name and also remove the query parameter from the original request. For example, you want to replace the query parameter `dept` with the parameter `department`. The API column should have the parameter to be replaced, and the Service column should have the replacement value for the parameter. If the query in the request is not `dept`, then the original query parameter is not replaced or removed. If there is a key value pair for example `key=value` in the API column, if the incoming request query parameter and its value match then only the query parameter is replaced or renamed.

API	Service
`dept`	`department`
`acno`	`AccountNo`
`dept=123`	`department=sales123`
`dept`	`department=${payload.conditions[0].headerName}`

- **Remove Query Parameter**: The incoming request has a query parameter, and you want to remove the query parameter from the original request. For example, you want to remove the parameter with the name `xyz` from the request. The API column should have the name of the query parameter, and the Service column should be blank.

API	Service
`xyz`	—
`xyz=123`	—

Field Mapping

Fields are payload elements. This method allows you to modify the payload fields.

- **Replace Field Value**: You can replace the field value by entering the field and its value in the API column and field and its value to be replaced in the Service column.

API	Service
`defaultCondition.routeToUrl=replace`	`defaultCondition.routeToUrl=http://myserver/sales/`
`Conditions[0].headerName=xyz`	`Conditions[0].headerName=aabbcc`

- **Replace Field Name**: You can replace the field name by entering field name in the API column and the field name to be replaced in the Service column.

API	Service
conditions[1].headerName	conditions[1].departmentName
defaultCondition.routeToUrl	defaultCondition.renameUrl

- **Remove Field**: You can remove a particular field from the incoming request payload by entering field path in the API column and leaving the Service column blank.

API	Service
conditions[1].headerName	—
defaultCondition.routeToUrl	—

- **Add a Field**: You can add a particular field to the incoming request payload by leaving the API column blank and entering the path of the field and its value in the Service column.

API	Service
—	defaultCondition.saleArea=APAC
—	conditions[1].headerName=xyzcompany

- **Add a Dynamic Field or Value**: You can add a particular field to the incoming request payload by leaving the API column blank and entering the path of the field and its value in the Service column. You can also assign the header or query parameter value to the field.

API	Service
—	defaultCondition.saleArea=${payload.field1.fieldvalue}
—	conditions[1].headerName=${headers.abcd}

Applying REST to SOAP Policies

Use the REST to SOAP to expose a SOAP service as a JSON REST service.

You can configure the REST to SOAP policy in one dialog box for both the request and the response pipeline.

The default configuration for the service is pre-populated from the WSDL file in a SOAP service. You can modify the REST to SOAP policy configuration for the request and the response. For the request, you can use dynamic path parameters to map resources to WSDL operations. THe SOAP payload can also be configured with dynamic values based on the REST request. For the response, the REST payload can be configured with dynamic values based on the SOAP response.

At runtime, the request part of the REST to SOAP policy is responsible for constructing the SOAP payload based on the JSON input. The response part of the policy is responsible for constructing the JSON response based on the SOAP output. All this is done without loading the provided WSDL/XSDs. Note that only JSON payloads are

supported. For SOAP faults, either system faults or business faults, the content is converted to JSON and sent to the REST caller.

Before you apply the REST to SOAP policy, you must create a SOAP service and upload a WSDL file. See Creating a Service for more information.

> **Note:**
>
> The policy configuration saved as part of the API does not include the content of the WSDL/XSDs.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a REST to SOAP policy:

1. In the Available Policies region, expand **Interface Management**, hover over **REST to SOAP**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. If you are editing the policy from the request flow, then from the **Place after the following policy in response pipeline** list, select the policy after which this policy is placed in the response flow. If you are editing the policy from the response flow, then from the **Place after the following in response pipeline** list, select the policy after which this policy is placed in the request flow.

   e. Click **Next**.

3. In the **Conversion Configuration** section, click **Select Service**.

4. Click an existing SOAP service with WSDL and click **Select**.

   All of the possible operations in the selected service are listed.

5. Click the box on the left of the row to select an operation. Enter the resource URL in the **Path** field. You do not need to select all the operations listed before applying the policy. When you edit the policy later, you are presented again with all the operations in the WSDL with the already selected operations pre-selected. If the WSDL was updated, the policy dialog box notifies you about the update and give you the option to synchronize the list by clicking a **Refresh** button.

   Click the down arrow on the right of the row to see a box in which you can view the REST to SOAP payload and the SOAP to REST payload.

6. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

7. When you click **Apply**, the Service Request policy is updated to be automatically configured with the REST to SOAP service selected above. Click **Ok** in the banner.

   Note: If the Service Request policy was already configured with either a service or a URL, the service policy is not updated. You should open the Service Request and choose the REST to SOAP service. Otherwise, you may receive an invalid API request message when the policy is invoked. If the Service Request policy was not yet configured, the REST to SOAP service is added automatically.

Applying Header-Based Routing Policies

Use the header-based routing policy to route incoming requests to a specific service request URL based on the presence or value of a specified header. For example, you can specify a different backend service to route requests to different credit rating agencies based on the value of a header you specify.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a header-based routing policy:

1. In the Available Policies region, expand **Routing**, hover over **Header Based Routing**, and then click **Apply**.

   The Edit Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. Enter a header name into the **Header** field.

   f. Choose an expression with which to evaluate the header. The following expressions are available:

      - **=** (is equal to)
      - **!=** (is not equal to)
      - **>** (is greater than)
      - **<** (is less than)
      - **>=** (is greater than or equal to)
      - **<=** (is less than or equal to)
      - **Is NULL** (the header is not present or has no value)
      - **Is NOT NULL** (the header is present and has a value)

g. Depending on the expression you selected, enter a relevant value into the **Value** field. Some expressions, like **IS NULL** and **IS NOT NULL**, do not require values.

h. **(Optional)** To drop or add/update headers, expand the **Configure Headers** section.

- To drop headers, click the **Drop Headers** field to list header names to be deleted for this condition. Click in the field and select a header to drop. Repeat for each header you want to drop. You can also type the header name in the field and press **Enter**.
- To add or update headers, click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field. Click the **+** (Add new Header) icon to add more headers as needed, specifying the header name and value for each one you add.

i. In the **Service** section, choose one of the following:

- **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.
- **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

  Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

j. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

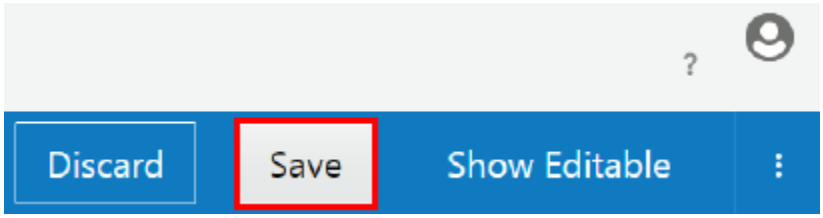
k. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2f through Step 2j to populate the data for any additional conditions you add.

l. In the **Otherwise** section, choose one of the following behaviors to route all requests not matching the conditions you specified above:

- **(Default)** Select **Keep Default Service Request URL** to route requests to the API's default service request URL.
- Select **Configure Service Request** to route requests to a specific service request. Configure headers, a service, and a service account as in the steps above.

m. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Gateway-Based Routing Policies

Use a gateway-based routing policy to route requests to different service request URLs based on the gateway to which the API is deployed.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a gateway-based routing policy:

1. In the Available Policies region, expand **Routing**, hover over **Gateway Based Routing**, and then click **Apply**.

   The Edit Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. In the **Gateways** list, select a gateway onto which the API is deployed that you want to route to a different service response URL, and then press **Enter**.

      You can enter multiple gateways. Press **Enter** after entering each gateway. Remove a gateway by clicking the **X** next to it.

   f. **(Optional)** To drop or add/update headers, expand the **Configure Headers** section.

      - To drop headers, click the **Drop Headers** field to list header names to be deleted for this condition. Click in the field and select a header to drop. Repeat for each header you want to drop. You can also type the header name in the field and press **Enter**.
      - To add or update headers, click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field. Click the **+**

(Add new Header) icon to add more headers as needed, specifying the header name and value for each one you add.

g. In the **Service** section, choose one of the following:

- **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.
- **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

  Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

h. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

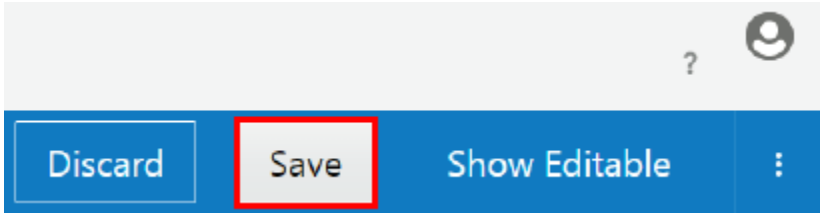
i. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Steps 2d through 2h to populate the data for any additional conditions you add.

j. In the **Otherwise** section, choose one of the following behaviors to route all requests not matching the conditions you specified above:

- **(Default)** Select **Keep Default Service Request URL** to route requests to the API's default service request URL.
- Select **Configure Service Request** to route requests to a specific service request. Configure headers, a service, and a service account as in the steps above.

k. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Application-Based Routing Policies

Use an application-based routing policy to route requests from specific applications or application types to a service request URL that you specify.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure an application-based routing policy:

1. In the Available Policies region, expand **Routing**, hover over **Application-Based Routing**, and then click **Apply**.

   The Edit Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select a key validation policy.

   > **Note:**
   >
   > The application-based routing policy must be used in tandem with a key validation policy.

   d. Click **Next**.

   e. In the **Applications** list, select the application from which you want to route requests to a different service response URL.

   You can enter multiple applications. Requests from all applications you enter in a single field are routed to the same Service URL. Press **Enter** after entering each application. Remove an application by clicking the **X** next to it.

   f. **(Optional)** To drop or add/update headers, expand the **Configure Headers** section.

   - To drop headers, click the **Drop Headers** field to list header names to be deleted for this condition. Click in the field and select a header to drop. Repeat for each header you want to drop. You can also type the header name in the field and press **Enter**.
   - To add or update headers, click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field. Click the **+** (Add new Header) icon to add more headers as needed, specifying the header name and value for each one you add.

   g. In the **Service** section, choose one of the following:

   - **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are

issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.

- **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

  Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

h. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

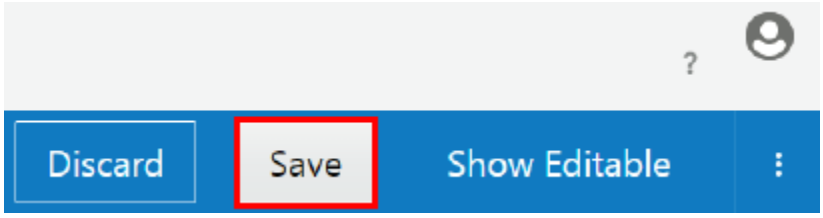
i. **(Optional)** Click the **+** (Add a new condition) icon to add additional application routing conditions. Repeat steps 2e through 2h to populate the data for any additional conditions you add.

j. In the **Otherwise** section, choose one of the following behaviors to route all requests not matching the conditions you specified above:

   - **(Default)** Select **Keep Default Service Request URL** to route requests to the API's default service request URL.
   - Select **Configure Service Request** to route requests to a specific service request. Configure headers, a service, and a service account as in the steps above.

k. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Resource-Based Routing Policies

Use the Resource-based routing policy to route requests to specific resource paths to different service request URLs.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a resource-based routing policy:

1. In the Available Policies region, expand **Routing**, hover over **Resource-Based Routing**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

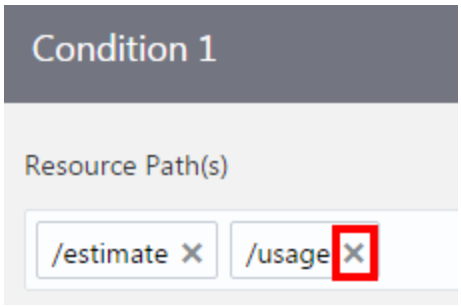
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. In the **Resource Paths** field, type the resource path(s) (like `/estimate`, including the slash) that you want to route to a different service request URL, and then press **Enter**.

      You can enter multiple resource paths. Press **Enter** after entering each resource path. Remove a path by clicking the **X** next to it.



   f. **(Optional)** To drop or add/update headers, expand the **Configure Headers** section.

      - To drop headers, click the **Drop Headers** field to list header names to be deleted for this condition. Click in the field and select a header to drop. Repeat for each header you want to drop. You can also type the header name in the field and press **Enter**.
      - To add or update headers, click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field. Click the **+**

(Add new Header) icon to add more headers as needed, specifying the header name and value for each one you add.

g. In the **Service** section, choose one of the following:

- **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.
- **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

  Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

h. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

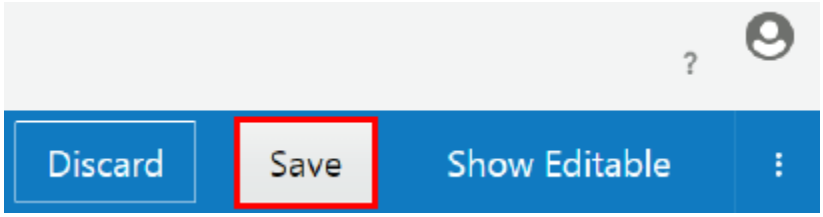
i. **(Optional)** Click the **+** (Add a new condition) icon to add additional conditions. Repeat Step 2e through Step 2h to populate the data for any additional conditions you add.

j. In the **Otherwise** section, choose one of the following behaviors to route all requests not matching the conditions you specified above:

- **(Default)** Select **Keep Default Service Request URL** to route requests to the API's default service request URL.
- Select **Configure Service Request** to route requests to a specific service request. Configure headers, a service, and a service account as in the steps above.

k. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Service Callout 2.0 Policies

Use a service callout policy to call external services from an API's request flow. The gateway can pass or reject the request based on the status code received from the external system.

You can use a service callout policy to create an object in another system during the request flow. The service callout policy can use GET, PUT, POST or DELETE methods to call external services.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a service callout policy:

1. In the Available Policies region, expand **Other**, hover over **Service Callout 2.0**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:
   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.
   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.
   d. Click **Next**.
   e. Select **PASS** to pass, or select **REJECT** to reject, requests that meet the criteria configured in this policy.
   f. Select **Specific** from the list, and then enter a status code into the **Status Codes** field that, when received, the request is passed or rejected as you specified above. You can enter multiple status codes in this field, separating each one by pressing **Enter**. Alternatively, select **Any** from the list to pass or reject the request if any status is received from the external service.
   g. **(Optional)** Select the **Include timeouts and exceptions as conditions** option to use timeouts and other exceptions received from the external system to pass or reject requests. Deselect this option if you want to ignore timeouts and other exceptions.
   h. From the **Method** list, select the method used to call the external service: **GET**, **POST**, **PUT**, or **DELETE**.
   i. In the **Service** section, choose one of the following:
      - **Select Existing**: Click the **Select Service** button to select a service from the list of services available. You can only add services for which you are issued the Manage Service or Reference Service grant. See Creating a Service and Issuing Service Grants.
      - **Enter a URL**: Use this option to enter a URL for the service. With this option, you can also select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

Note: Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

j. **(Optional)** Click the **Select Service Account** button and select the service account containing credentials required to access the service. Note that selecting a service account here overrides any service account attached to the service, if you selected one above.

   You can only add service accounts for which you are issued the Manage Service Account or Reference Service Account grant. See Creating a Service Account and Understanding Service Account Grants.

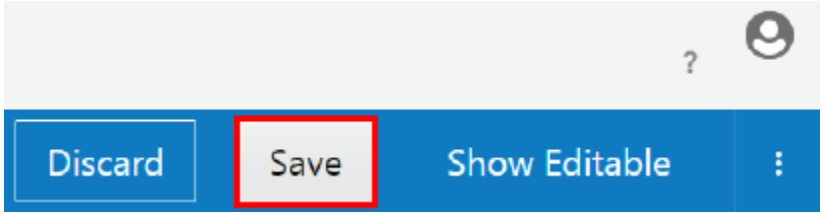
k. **(Optional)** To add or update headers, expand the **Add/Update Headers** section. Click the **Header Name** list and select the header you want to add or update, or type the header name and press **Enter**. Enter a value for the header in the **Header Value** field.

   Click the **+** (Add new Header) icon to add another header as desired and repeat this step for additional headers you want to send to the service. The policy supports passing the inbound headers with ${headers.headername}. For more information, see About Using Groovy in Policies

l. **(Optional)** In the **Request Payload** field, enter the request payload to send to the external service you want to call, if applicable. This payload can be JSON, XML, or another content type accepted by the service. This field is only available if you selected POST or PUT as the method in Step 2h.

m. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Service Callout 1.0 Policies

Use a service callout policy to call external services from an API's request flow. The gateway can pass or reject the request based on the status code received from the external system.

You can also use a service callout policy to create an object in another system during the request flow. The service callout policy can use GET, PUT, POST or DELETE methods to call external services.

This policy can be added only to the request flow.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a service callout policy:

1. In the Available Policies region, expand **Other**, hover over **Service Callout 1.0**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request flow.

   d. Click **Next**.

   e. Select **PASS** to pass, or select **REJECT** to reject, requests that meet the criteria configured in this policy.

   f. Select **Specific** from the list, and then enter a status code into the **Status Codes** field that, when received, the request is passed or rejected as you specified above. You can enter multiple status codes in this field, separating each one by pressing **Enter**. Alternatively, select **Any** from the list to pass or reject the request if any status is received from the external service.

   g. **(Optional)** Select the **Include timeouts and exceptions as conditions** option to use timeouts and other exceptions received from the external system to pass or reject requests. Deselect this option if you want to ignore timeouts and other exceptions.

   h. From the **Method** list, select the method used to call the external service: **GET**, **POST**, **PUT**, or **DELETE**.

   i. In the **Service URL** field, enter the URL of the service you want to call.

   j. **(Optional)** Select the **Use Gateway Node Proxy** option if a proxy is required to call the service from the gateway node your API will be deployed to.

      Gateway Managers with the Manage Gateway grant can configure proxies for gateway nodes they manage. This allows for different proxy configurations for dev and production gateways or different proxies for nodes located in separate data centers. If this option is selected, but no proxy is configured for a node the API is deployed to, the request is passed from the gateway node to the backend service without using a proxy. See Configuring a Proxy for a Gateway Node.

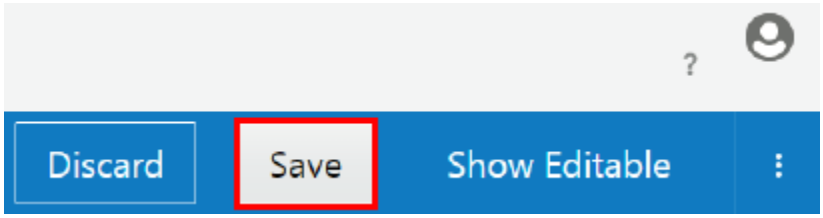
   k. **(Optional)** In the **Headers** section, click the **+** (Add a new header icon), and then enter a header key:value pair to send to the service in the following format: `Content-Type:application/json`.

      Repeat this step for additional headers you want to send to the service.

   l. **(Optional)** In the **Request Payload** field, enter the request payload to send to the external service you want to call, if applicable. This payload can be JSON, XML, or another content type accepted by the service. This field is only available if you selected POST or PUT as the method in Step 2h.

m. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Applying Groovy Script Policies

Use a Groovy Script policy to pass or reject a request by examining the request context or to manipulate the request context.

This policy can be added to the request or response flows.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a Groovy script policy:

1. In the Available Policies region, expand **Other**, hover over **Groovy Script**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

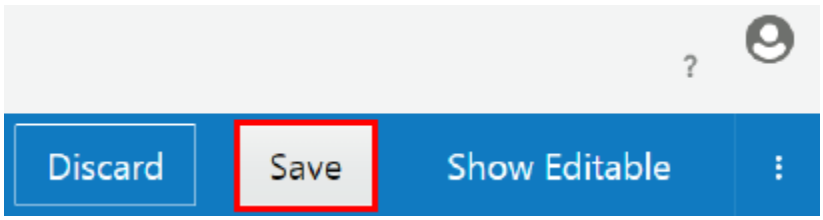
   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request or response flow.

   d. Click **Next**.

   e. In the **Groovy Script** field, enter the Groovy script that you want to run when a request reaches this point in the request or response flows.

   f. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Example 4-2 Example of Header Validation Policy Configuration in JSON

The policy configuration would be part of an API configuration.

```
{
    type: "o:GroovyScript",
    name: "Groovy Script",
    version: "1.0",
    category: "@implementations.policyCategory.other",
    description: "Executes Groovy script",
    constraints: {
        direction: "REQUEST_OR_RESPONSE",
        singleton: false
    },
    ui: {
        edit: {
            html: "groovyscript-edit.html",
            js: "groovyscript-edit.js",
            helpInfo: "#helpInfo",
            helpUrl: "http://www.oracle.com",
            helpTopicId: ""
        },
        view: {
            html: "groovyscript-view.html",
            js: "groovyscript-view.js",
            helpInfo: "#helpInfo",
            helpUrl: "http://www.oracle.com",
            helpTopicId: ""
        },
        l10nbundle: "groovyscript.js"
    }
}
```

Example 4-3 Groovy Script Action Configuration in XML

This is the XML configuration that is required by OCSG gateway to configure the policy.

```
def xmlText1 = '''<?xml version="1.0" encoding="UTF-8" ?>
<OUT_PUT>
<id>12345678</id>
<name>freedom</name>
<email />
</OUT_PUT>'''
context.apiResponse.setBody(new StringBodyImpl(xmlText1, null))
```

Applying Logging Policies

Use a logging policy to log and store custom messages for APIs deployed to a specific gateway.

This policy can be added to the request or response flows.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

To configure a logging policy:

1. In the Available Policies region, expand **Other**, hover over **Logging**, and then click **Apply**.

   The Apply Policy dialog appears.

2. From the Apply Policy dialog:

   a. **(Optional)** In the **Your Policy Name** field, enter a name for the policy.

   b. **(Optional)** In the **Comments** field, describe why you are applying the policy for this API.

   c. From the **Place after the following policy** list, select the policy after which this policy is placed in the request or response flow.

   d. Click **Next**.

   e. In the **Log Level** list, select the level at which this message is logged: **Info**, **Warning**, **Error**, or **Severe**.

   f. In the **Log Message** field, enter the message to be written to the log. Using Groovy notation, you can log values of headers, fields, or other values in the outbound request or inbound response. For example, to log the value of a `tenant-id` header sent with requests, enter `tenant-id is ${headers.tenant-id}`. See About Using Groovy in Policies to learn about using Groovy notation with this policy.

      Using this example, when a `tenant-id` header with a value of `2` is sent with a request, this line is written to the log file you specify in the next step:

      ```
      [06-28 01:00:55: logging.LoggingValidationAction][[ACTIVE]
      ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-
      tuning)'] INFO  oracle.apiplatform.customlog  :  tenant-id is 2.
      ```

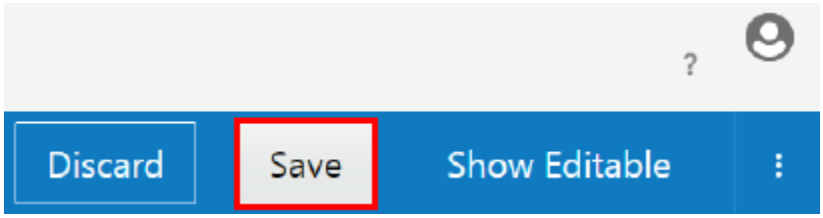
      A line is written for each request or response that reaches the Logging policy in the flow. Requests or responses rejected before the Logging policy executes are not logged. If a request or response does not include the header or field you specify (it is null or hidden by the Redaction policy), a line like this is written to the log:

      ```
      [06-28 01:17:50: logging.LoggingValidationAction][[ACTIVE]
      ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-
      tuning)'] INFO  oracle.apiplatform.customlog  :  tenant-id is '
      Failed to find: ${headers.tenant-id} '.
      ```

   g. In the **Log File Path** field, enter the location to the log file you want to write messages to. This path is relative to this path on the gateway node domain: `domains/<name of the gateway domain>/apics/customlog/<test/ora.log>`. For example, if you enter `api.log`, events are written to `domains/<name of the gateway domain>/apics/customlog/api.log`.

h. Click **Apply** to save your changes and close the dialog, or click **Apply as Draft** to save a draft of your policy configuration. See Working with Draft Policies.

3. Click **Save**.



The policy is now added to the API. It is activated when the API is (re)deployed to a gateway.

Working with Draft Policies

You can save a draft policy if you want to save your work but have not yet completed configuring a policy.

Draft policies are not activated when an API is deployed to a gateway.

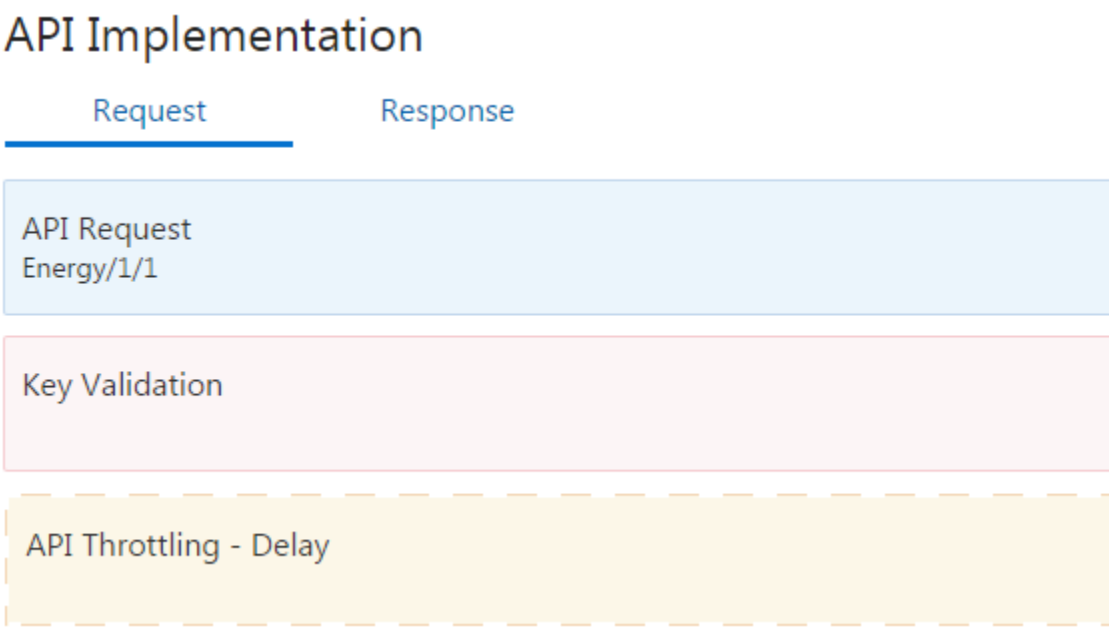
You can save a draft policy with validation errors; you must fix these errors before applying the policy and deploying the API to a gateway. The next time the API is deployed, the policies you've applied are activated.

This task assumes that you are already viewing the API Implementation tab for an API. To navigate to this tab, first open an API from the APIs tab, and then click the **API Implementation** tab.

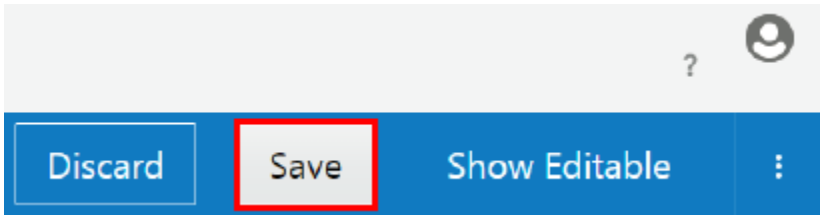
To apply and deploy a draft policy:

1. Hover over the draft policy you want to apply and click **Edit**.

   On the API Implementations tab, draft policies are displayed with a dotted-line border.



2. Make any changes that you want to the policy configuration.
3. Click **Apply**.
4. Click **Save**.



5. Deploy or redeploy the API, as described in Deploying or Redeploying an API Endpoint to a Gateway.

About Using Groovy in Policies

You can use Groovy notation to access dynamic contents like headers, query parameters, payloads and some context properties in specific policies.

These policies include:

- Method Mapping
- Service Callout
- Logging
- Redaction
- Groovy Script

Reserved Top-Level Variables

The following variables are reserved for system use and can be used in any policy that supports Groovy scripting:

- `headers`: a map of incoming headers. Access a specific header with this syntax: `${headers.nameOfHeader}`.
- `queries`: a map of incoming query parameters. Access a specific query parameter with this syntax: `${queries.nameOfQueryParam}`.
- `payload`: the parsed XML or JSON payload. Access an element in the payload with this syntax: `${payload.invoice.quote}`
- `msgProperties`: a map of context properties. This variable is reserved but not implemented.

> **Note:**
>
> If the value of any of the dynamic entries are empty, `null` is used in place of the empty value. For example, if a policy accesses a header named tenant-id using the Groovy notation `${headers.tenant-id}`, but the header is absent or has no value, this notation is evaluated as `null`.

Example 4-4 Creating a New Query Parameter with the Method Mapping Policy

The following example maps an incoming query parameter, `abc`, to a new query parameter, `xyz`. Instead of using the value of `abc`, the value of `xyz` will be the value of `${payload.invoice.quote}` in the incoming payload.

Example 4-5 Creating a New Header with the Method Mapping Policy

The following example maps an incoming header, `abc`, to a new header, `xyz`. Instead of using the value of `abc`, the value of `xyz` will be the value of `${headers.abc}` in the incoming payload.

Example 4-6 Constructing an XML Payload with the Service Callout Policy

The following example creates an XML payload using values from the outbound request sent to a deployed endpoint. The value of the `customer.name` field is mapped to a `<user>` XML element in the payload sent to an external service; the value of the `customer.age` field is mapped to an `<age>` element in the payload.

```
<output><user>${payload.customer.name}</user><age>${payload.customer.age}</
age></output>
```

Example 4-7 Constructing a JSON Payload with the Service Callout Policy

The following example creates a JSON payload using values from the outbound request sent to a deployed endpoint. The value of the `customer.name` field is mapped to a `user` JSON object in the payload sent to an external service; the value of the `customer.age` field is mapped to an `age` object in the payload.

```
{"user":"${payload.customer.name}", "age":${payload.customer.age}}
```

Example 4-8 Logging Dynamic Values with the Logging Policy

The following example writes an entry to the log file every time the policy is triggered, replacing the Groovy variables with values from the introspected payload:

```
Got an order from ${payload.user}, the amount is ${payload.price}.
```

Deploying Endpoints

API Managers can use the Management Portal to deploy, redeploy, or undeploy API endpoints to gateways, if issued the required grants.

Topics

Deploying or Redeploying an API Endpoint to a Gateway

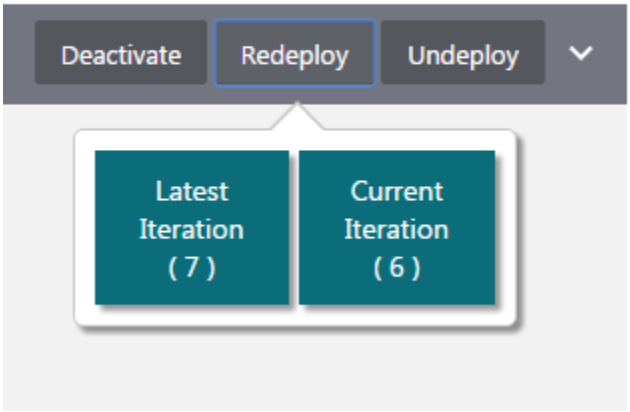
Deploy an endpoint for your API to a gateway when you're ready for it to receive requests.

To deploy an endpoint, API Managers must have the Manage API or Deploy API grant for the API in addition to the Deploy to Gateway or Request Deployment to Gateway

grant for a gateway. If they have the Request Deployment to Gateway grant, the request must be approved by a Gateway Manager before the endpoint is deployed; it remains in a Requesting state until it is approved or rejected. If they have the Deploy to Gateway grant the request is automatically approved.

1. From the APIs List page, select the API that you want to deploy.
2. Click the (Deployments) tab.
3. To deploy an API that is not already deployed to the gateway:
   a. Click **Deploy API**.
   b. Use the Filter field to find and select the gateway you want to deploy to.
   c. From the Initial Deployment State section, select **Active** to deploy the API in an active state, or select **Inactive** to deploy the API in an inactive state.
   d. (Optional) In the **Description** field, enter comments about the API deployment.
   e. Click **Deploy**.

   The deployment enters a **Waiting** state and the logical gateway definition is updated. The endpoint is deployed the next time gateway node(s) poll the management server for the updated gateway definition.
4. To redeploy an API:
   a. Hover over the Production Gateway deployment, and click **Redeploy** when it appears.
   b. Click **Latest Iteration** to deploy the most recently saved iteration of the API, or click **Current Iteration** to redeploy the currently deployed iteration of the API.



   c. When prompted, enter comments about why you are redeploying the API, and then click **Yes**.

   The deployment enters a **Waiting** state and the logical gateway definition is updated. The endpoint is deployed the next time gateway node(s) poll the management server for the updated gateway definition.

The deployment request is submitted. Depending on the grant combinations you are issued, the request might have to be approved by a Gateway Manager user.

The endpoint moves to the **Deployed** tab when the deployment is successful and approved, if applicable.

Undeploying an API from a Gateway

Undeploy an API if you no longer want gateway nodes to process requests for it.

API Managers must be issued the Manage API grant for the API and the Deploy to Gateway or Request Deploy to Gateway grant for the gateway to undeploy that API from that gateway. If you have the Request Deploy to gateway grant, a Gateway Manager must approve the undeployment request.

To undeploy an API from a gateway:

1. From the APIs List page, select the API you want to undeploy.
2. Click the  (Deployments) tab.
3. Hover over the API you want to undeploy, and then click **Undeploy**.
4. When prompted, enter comments about why you are redeploying the API, and then click **Yes**.

The undeployment request enters the **Waiting** state, which means that the undeployment request is pending. The API is undeployed from nodes registered to the gateway when each polls the management service for the latest logical gateway definition.

Managing API Grants

API grants allow you to issue fine-grained permissions to users or groups for specific APIs.

Topics

Understanding API Grants

API grants are issued per API.

Users and groups issued grants for a specific API have the privileges to perform the associated actions on that API. See Issuing API Grants to issue API grants.

Grant Name	Description	Can Be Issued To	Associated Actions
Manage API	People issued this grant are allowed to modify the definition of and issue grants for this API.	API Managers	APIDelete APIViewAllDetails APIViewPublicDetails APIEdit APIEditPublic APIModifyPublishState APIModifyLifecycleState APIDeploy APIGrantManageAPI APIGrantViewAllDetails APIGrantViewPublicDetails APIGrantDeployAPI
View all details	People issued this grant are allowed to view all information about this API in the Management Portal.	API Managers, Gateway Managers, Plan Managers	APIViewAllDetails
View public details	People issued this grant are allowed to view the publicly available details of this API on the Developer Portal. This grant can be issued to users of any role.	API Managers, Application Developers, Plan Managers	APIViewPublicDetails
Entitle API	Users issued this grant are allowed to entitle this API to a plan for which they have entitle rights.	API Managers, Plan Managers	APIEntitlementAdd APIEntitlementEdit APIEntitlementRemove APIEntitlementModifyState APIEntitlementModifyPublishState
Deploy API	API Managers with the Manage API grant already have this permission for all gateways they are allowed to view. API Managers without the Manage API grant and Gateway Managers issued this grant are allowed to deploy or undeploy this API to a gateway for which they have deploy rights. This allows Gateway Managers to deploy this API without first receiving a request from an API Manager.	API Managers, Gateway Managers	APIDeploy

Issuing API Grants

Issue API grants to users or groups to determine what actions assignees can perform with that API. See Understanding API Grants. Grants are issued per API; repeat this task for each API you want to issue grants for.

You must be issued the Manage API grant for an API to issue grants for it.

1. On the APIs List page, select the API for which you want to manage grants.
2. Click the  (Grants) tab.
3. Click the tab that corresponds to the grant you want to issue to users or groups:
   - **Manage API**: API Manager users issued this grant are allowed to modify the definition of and issue grants for this API.
   - **View all details**: API, Gateway, and Plan Manager users issued this grant are allowed to view all information about this API in the Management Portal.
   - **Deploy API**: Gateway Managers issued this grant are allowed to deploy or undeploy this API to a gateway for which they have deploy rights. This allows Gateway Managers to deploy this API without first receiving a request from an API Manager. API Managers already have this permission due to the Manage API grant. API Managers issue this grant to Gateway Managers for APIs that they own.
   - **Entitle API**: Users issued this grant are allowed to entitle this API to a plan for which they have entitle rights.
   - **View public details**: Users issued this grant are allowed to view the publicly available details of this API on the Developer Portal. This grant can be issued to users of any role.
4. Click **Add Grantee**.

   The Add Grantee dialog appears.
5. From the Add Grantee dialog, select the user(s) or group(s) to which you want to issue the grant. You can select multiple users and groups.

   > **Note:**
   >
   > You cannot select users or groups that already have this grant; they are greyed out in the Add Grantee dialog.
6. Click **Add**.

The user(s) or group(s) are issued the API grant you chose.

Managing API Entitlements

An entitlement is the relationship between an API and a plan. Review the topics to manage API entitlements.

Topics:

Understanding API Entitlements

An entitlement is the relationship between an API and a Plan that defines how a client application can access the API.

There is a many-to-many relationship between plans and APIs. A given plan can have entitlements to multiple APIs; for example, to group related APIs. A given API can be entitled by multiple plans; for example, to provide different quality of service criteria. Note that two entitlements of the same plan cannot point to the same entire API, or to the same action in an API.

Entitlement through a plan is not required to invoke an API. You can decide how you want the APIs to be exposed.

- **Keep API open**: The API does not require plans nor applications.
- **Unmanaged**: A plan is not required, but the client needs to create an application to access API.
- **Managed**: The API is entitled to one or more plans, and the client must create an application and subscribe that application to one of the plans.

Viewing API Entitlement Details

You can view the entitlements that are added to the API and other details such as whether the entitlement is active or inactive and whether it is published or unpublished.

To view details of an entitlement:

1. On the APIs List page, click the API for which you want to view the entitlements.
2. Click the  (Entitlements) tab.
3. To filter the list for active or inactive entitlements, click the **Active** or **Inactive** tab.
4. To filter the list for published or unpublished entitlements, click the **Published** or **Unpublished** tab.
5. To see all entitlements, click the **All** tab.

6. To view more details about an entitlement, click its name or the **Expand** icon to the right.

   The plan ID and description appear.

Adding an Entitlement to an API

You can add an entitlement to an API from the API component.

To add entitlement to an API:

1. On the APIs List page, click the API to which you want to add an entitlement.
2. Click the  (Entitlements) tab.
3. Click **Add Entitlement**.
4. Select the plan in the **Add Plan** window that appears.

   You can add entitlements to multiple plans.
5. Choose either **Active** or **Inactive** to set the initial state of the entitlement.
6. Click **Add**.

Publishing and Unpublishing an Entitlement in an API

You must publish an entitlement in an API to enable the application developer to access it from Developer Portal.

To publish or unpublish an entitlement in an API:

1. On the APIs List page, click the API that you want to publish or unpublish.
2. Click the  (Entitlements) tab.
3. Click the name of the entitlement you want to publish or unpublish.
4. Click the **Publish** or the **Unpublish** button that appears.

   These buttons interchange depending on the publication state of the entitlement.
5. Click **Yes**.

Activating and Deactivating an Entitlement in an API

You must activate the entitlement in an API for the application developer to access it from developer portal.

To activate or deactivate an entitlement in an API:

1. On the APIs List page, click the API for which you want to activate or deactivate an entitlement.
2. Click the  (Entitlements) tab.
3. Click the name of the entitlement you want active or deactivate.
4. Click the **Activate** or the **Deactivate** button that appears.

These buttons interchange depending on the publication state of the entitlement.

5. Click **Yes**.

Removing an Entitlement from an API

You can remove an entitlement from the API that is not required.

To remove an entitlement from an API:

1. On the APIs List page, click the API from which you want to remove an entitlement.
2. Click the  (Entitlements) tab.
3. Click the name of the entitlement you want to remove.
4. Click the **Remove** button.
5. Click **Yes**.

Publishing APIs

Use the Publication page to publish APIs to the Developer Portal. On this page you provide general details about and provide detailed documentation references for your API. After publication, Application Developers use the Developer Portal to discover, evaluate, register, and consume your published APIs.

Topics

Configuring an API's Developer Portal URL

Before publishing to the Developer Portal, each API has to be configured with its own unique **Vanity Name**. A vanity name is the URI path of an API's details page when it is published to the Developer Portal.

For example, if your Developer Portal URL is `http://example.com:7201/developers/apis/`, you can publish the details page for your API to `http://example.com:7201/developers/apis/Energy`, where `Energy` is the vanity name you enter in the Management Portal.

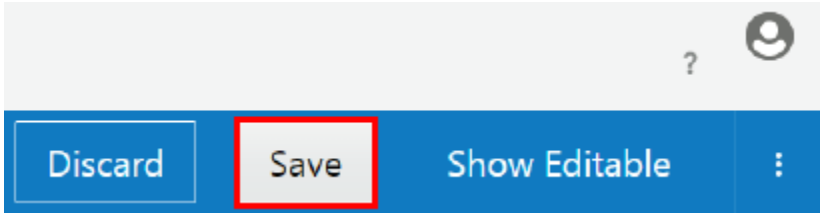
To configure an API's Developer Portal URL:

1. From the APIs List page, select the name of the API that you want to edit.
2. On the **Publication** tab, enter the path at which this API will be discoverable in the Developer Portal in the **API Portal URL** field. This is also called the API's **vanity name**.

   For example, enter `creditcheck`. When published, the details page for this API is visible at `https://<host>:<port>/developers/apis/creditcheck`.

> **Note:**
>
> An API's vanity name must be unique, regardless of case. You can't have APIs with vanity names of `Creditcheck` and `creditcheck`. You must enter the vanity name **exactly** (matching case) in the URL to navigate to an API's details page in the Developer Portal. For example, navigating to `https://<host>:<port>/developers/apis/Creditcheck` opens the page for an API with a vanity name of `Creditcheck`; `https://<host>:<port>/developers/apis/creditcheck` doesn't open this page and returns a 404 because the segment in the URL does not match the vanity name exactly.
>
> Only valid URI simple path names are supported. Characters such as "?", "/", and "&" are not supported in vanity names. `Test_2` is a supported vanity name, but `Test/2` is not.

3. Click **Save**.



Before publishing an API, you should also configure it's overview text and documentation references. See Adding Overview Text for an API and Documenting an API.

The API is not visible on the Developer Portal until you publish it. See Publishing an API to the Developer Portal.

Adding Overview Text for an API

You can provide overview text for an API, describing its features and other information a developer should know about its use, in either HTML or Markdown.

You can upload a file, enter text manually, or provide a link to HTML or Markdown to use as overview text. This text appears on the API's detail page in the Developer Portal.

> **Note:**
>
> Detailed use information for an API, such as an API's resources and methods, are better described in documentation references. See Documenting an API.

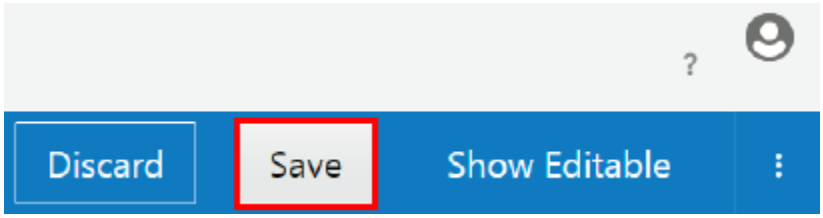
To add overview text for an API:

1. On the APIs List page, select the API to which you want to add overview text.
2. Click the Publication tab.

3. From the **Developer Portal API Overview** section, click **HTML** or **Markdown**, depending on the format of your overview text.
4. Do one of the following:
   a. To add overview text stored in a file, click the **File** tab, and then click **Choose File**. Navigate to and select the file on your local disk that contains the overview text you want to display. After selecting your file and closing the file browser, Click **OK**.
   b. To manually enter overview text, click the **Text** tab, and then enter HTML or Markdown overview text. Click **OK**.
   c. To display overview text from a web page, click the **Link** tab, and then enter the URL of the overview text page you want to display. Click **OK**.

> **Note:**
>
> The entire web page at the URL you enter appears in an iframe in the Developer Portal.

5. Click **Save**.



The overview text is added to the API's detail page in the Developer Portal. You must (re)publish this API to the Developer Portal before this text is visible. See Publishing an API to the Developer Portal.

Documenting an API

Publishing an API allows application developers to discover and register applications to the API. API publication and deployment are two separate activities; API publication allows consumers with the correct grants to access the API web page and API deployment makes the API endpoint accessible. Use the topics in this section to learn more about publishing API details to the Developer Portal.

> **Important:**
>
> The API Request URL, configured in Configuring the API Request URL, is not displayed in the Developer Portal. You must list it in the API's documentation so Application Developers know where to send requests.

Topics

- Adding HTML, Markdown, or Web Page Documentation to an API
- Adding Oracle Apiary Documentation to an API

Adding HTML, Markdown, or Web Page Documentation to an API

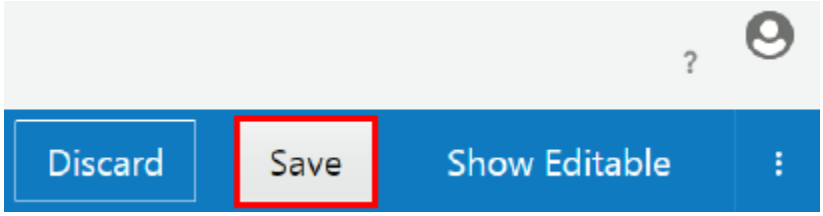
You can provide HTML or Markdown documentation by uploading a file, manually entering text, or providing a URL to the documentation resource. After you have added the documentation, it appears on the **Documentation** tab of the API detail page in the Developer Portal.

To add HTML, Markdown, or web page documentation to an API:

1. From the APIs List page, select the API to which you want to add documentation.
2. Click the **Publication** tab.
3. From the **Documentation** section, click **HTML** or **Markdown**, depending on the format of your documentation. See Adding Oracle Apiary Documentation to an API if you want to add documentation from Oracle Apiary.
4. Do one of the following:
   a. To add documentation stored in a file, click the **File** tab, and then click **Choose File**. Navigate to and select the file on your local disk that contains the documentation text you want to display. After selecting your file and closing the file browser, Click **OK**.
   b. To manually enter documentation text, click the **Text** tab, and then enter HTML or Markdown text. Click **OK**.
   c. To display documentation text from a web page, click the **Link** tab, and then enter the URL of the overview text page you want to display. Click **OK**.

   > **Note:**
   >
   > The entire web page at the URL you enter appears in an iframe in the Developer Portal.

5. Click **Save**.



Adding Oracle Apiary Documentation to an API

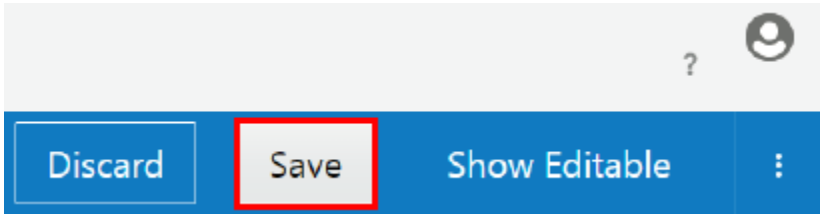
Use this procedure to add Oracle Apiary documentation to an API. Adding documentation to the API can help users understand its purpose and how it was configured.

> **Note:**
>
> Swagger or API Blueprint documentation can only be added to an Oracle Apiary Pro account. To add documentation, the team must have ownership of the API in Oracle Apiary. API definitions owned by personal accounts cannot be used. To transfer ownership of an API from a personal account to a team account, see the Oracle Apiary documentation.

To add Oracle Apiary documentation to an API:

1. On the APIs List page, select an API.
2. Click the **Publication** tab.
3. Click the **Apiary** button.

   The Apiary Documentation dialog appears, allowing you to browse documentation on Oracle Apiary.
4. Select an API Project and then click **Connect**.
5. Click **Save**.



If you have previously published the API, you must republish it to see the Apiary specification.

Publishing an API to the Developer Portal

Publish an API to the Developer Portal when you want application developers to discover and consume it.

Each published API has a details page on the Developer Portal. This page displays basic information about the API, an overview describing the purpose of the API, and documentation for using the API. This page is not visible on the Developer Portal until you publish it.

To publish an API to the Developer Portal:

1. From the API List page, click the name of the API you want to publish.
2. Click the **Publication** tab.

3. Ensure that the vanity name is present in the **API Portal URL** field. See Configuring an API's Developer Portal URL if you haven't entered a vanity name yet. Click **Save** if you have made any changes on this page.
4. **(Recommended)** Ensure that you've added overview text and documentation references. You can publish an API without providing these, but Application Developers need this information to know how to use your APIs.

   If you haven't added these yet, see Adding Overview Text for an API and Documenting an API.

   > **Important:**
   >
   > The API Request URL, configured in Configuring the API Request URL, is not displayed in the Developer Portal. You must list it in the API's documentation so Application Developers know where to send requests.

5. Click **Publish to Portal**.

The API is now visible on the Developer Portal. You can view its details page at the URL displayed in the **API Portal URL** field or by clicking the  (**Launch Developer Portal in another browser window**) icon next to the URL.

> **Note:**
>
> The HTML documentation is embedded in an iframe in the Developer Portal. Due to security constraints, a few browsers do not let an HTTP frame to be embedded into an HTTPs frame. They neither load the content nor give an error message. If the content does not show, change the `http://` in the address bar to `https://` and reload. The content displays correctly.

Delete an API

Administrators and API Managers can delete APIs in the Management Portal.

You can't delete an API if it is currently deployed to a gateway or if you don't have the Manage API grant for the API. Ensure you undeploy it from all gateways and that you have the proper grant before trying again.

To delete an API:

1. On the APIs List page, select the API you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes** in the banner to confirm.

The API is deleted.

5

Managing Services and Service Accounts

Service accounts and services are resources that can be easily referenced by policies in an API. The service account stores credentials for outbound authentication in a policy, and the service represents the backend service.

Topics

Managing Service Accounts

A service account externalizes credentials so that any policy with outbound calls or routing can reference it easily.

This chapter describes how to create and edit service accounts and manage grants to service accounts.

Topics

Typical Workflow for Managing Service Accounts

To start using service accounts, refer to the typical task workflow.

Task	Description	More Information
Create service accounts	Create service accounts in the Management Portal.	Creating a Service Account
Issue grants	Issue grants to control access to the service accounts.	Issuing Service Account Grants

Task	Description	More Information
Use service accounts in policies	Use the service accounts when you apply policies to control access to APIs.	Configuring the Service Request URL Applying Service Callout 2.0 Policies Applying Header-Based Routing Policies Applying Gateway-Based Routing Policies Applying Application-Based Routing Policies Applying Resource-Based Routing Policies

What Is a Service Account?

A service account is a resource containing credentials. Policies using outbound calls or routing can reference this resource to provide the necessary credentials.

You can use two authentication schemes with service accounts, Basic Auth and OAuth. Basic Auth has only two properties, username and password. OAuth has the following properties:

- **Token Endpoint URL**: The OAuth Token Provider endpoint where the access token is available.
- **Scope**: The scope(s) of the access request
- **Client ID**: The ID which identifies the client application.
- **Client Secret**: The secret password associated with the client ID. See Introduction to OAuth for more information.
- **Grant Type**: Either Client Credentials or Resource Owner Password Credentials. If you choose the Resource Owner Password Credentials, you must supply the appropriate Username and Password.
- **Token Transfer**: Transfer the token via URL or Header.

Understanding the Service Account List Page

The Service Accounts List page displays all service accounts created in the Management Portal.

Entries for service accounts display the following information:

- The name of the service account.
- The account type, either Basic Auth or OAuth 2.0.
- The date and time the service account was created and which user created it.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.

- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even within words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

> **Note:**
>
> If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Creating a Service Account

Create an entry for a service account you want to manage in the Oracle API Platform Cloud Service - Classic Management Portal.

To create a service account:

1. From the Service Accounts page, click **Create Service Account**.
2. In the **Service Account Name** field, enter the name of the service account. Note that the name of the service account should be unique.
3. (Optional) In the **Description** field, enter a brief description of the service account.
4. From the **Account Type** list, select either **Basic Auth** or **OAuth 2.0**.
5. If you selected **Basic Auth**, do the following.
   a. In the **Username** field, enter the user name.
   b. In the **Password** field, enter the password.

   > **Note:**
   >
   > The **Show Password** slider allows you to see the password or display asterisks.

   c. Click **Create**.
6. If you selected OAuth 2.0, do the following.

a. In the **Token Endpoint URL** field, enter the URL for the OAuth token provider endpoint where the access token is available.

b. (Optional) Click **Use Gateway Node Proxy** if a proxy is required to reach the token endpoint URL.

c. In the **Scope** field, enter a scope, such as `.READ`. Separate multiple scopes with a blank space.

d. In the Client ID field, enter the client ID.

e. In the Client Secret field, enter the client secret.

> **Note:**
>
> The **Show Client Secret** slider allows you to see the client secret or display asterisks.

f. From the **Grant Type** list, select **Client Credentials** or **Resource Owner Password Credentials**. If you select **Resource Owner Password Credentials**, enter the appropriate username and password.

g. In the **Token Transfer** section, click **Pass Token via URL** or **Pass Token via Header**.

h. Click **Create**.

The service account is displayed on the Service Account page.

Viewing Service Account Details

You can view the details of an API in a side panel available from any of the tabs.

The side panel displays the following details:

- The name of the service account.
- The description of the service account.
- The account type: **Basic** or **OAuth**.
- The most recent date and time that changes were saved for the service account and the name of the user who saved them.

To view service account details:

1. On the Service Account List page, select the service account for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the service account.

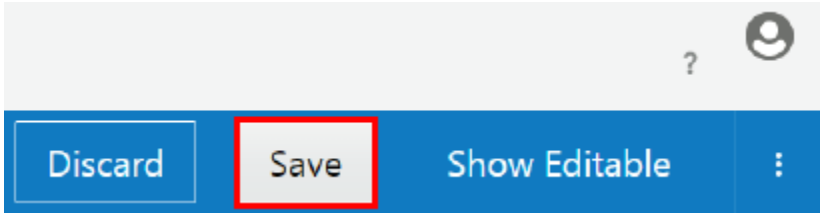
Editing Service Account Details

After you create a service account, you can then edit the details, including changing passwords.

To edit service account details:

1. From the Service Accounts List page, click the service account you want to edit.
2. Click the drawer icon to display the side panel.

3. Edit the name of the service account or the description in the side panel.
4. Click **Save**.



Deleting a Service Account

You can delete a service account at any time.

To delete a service account:

1. On the Service Accounts List page, click the service account you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes** in the banner to confirm.

Managing Service Account Grants

Service account grants allow you to issue fine-grained permissions to users or groups for each service account.

Topics

Understanding Service Account Grants

Service Account grants are issues per Service Account.

Users and groups issued grants for a specific Service Account have the privileges to perform the associated actions on that Service Account. See Issuing Service Account Grants to issue Service Account grants.

Grant Name	Description	Can be Issued To	Associated Actions
Manage Service Account	People issued this grant are allowed to view, modify and delete this service account.	Service Managers	ServiceAccountEditAll ServiceAccountViewAllDetails ServiceAccountViewHistory ServiceAccountDelete ServiceAccountReference ServiceAccountGrantManageServiceAccount ServiceAccountGrantViewAllDetails ServiceAccountGrantReferenceServiceAccount
View all details	People issued this grant are allowed to see all details about this service account.	API Managers, Gateway Managers, Service Managers	ServiceAccountViewHistory ServiceAccountViewAllDetails
Reference Service Account	People issued this grant are allowed to reference this service account (add it to policies).	API Managers, Service Managers	ServiceAccountViewAllDetails ServiceAccountReference

Issuing Service Account Grants

Grants allow you to control access to service accounts.

To issue grants:

1. From the Service Accounts page, click the service account to which you want to issue grants.
2. Click the  (User Management) tab.
3. Click the tab of the grant type you want to issue.
4. Click the **Add Grantee** button.
5. Select the user or group from the **Add to Grant** list. You can select multiple users.
6. Click **Add**.

Managing Services

A service resource allows you to store a backend URL that can then be referenced easily by policies.

Topics

Typical Workflow for Managing Services

To start using services, refer to the typical task workflow.

Task	Description	More Information
Create services	Create services in the Management Portal.	Creating a Service
Issue grants	Issue grants to control access to the services.	Issuing Service Grants
Use services in policies	Use the services when you apply policies to control access to APIs.	Configuring the Service Request URL Applying Service Callout 2.0 Policies Applying Header-Based Routing Policies Applying Gateway-Based Routing Policies Applying Application-Based Routing Policies Applying Resource-Based Routing Policies

What is a Service?

A service is a resource that represents the backend service for an API.

For an API, you can either configure the backend service explicitly, also referred to as inline, or by referencing a service resource. Using service resources allows you to configure a backend service once and then use it for any policy. This also makes updating a backend service easier.

When you create a service, a name and a service URL are required. A version and a description are optional and can be added at any point. If a gateway node needs to

use a proxy to reach the service, you can use the gateway node proxy option. Note that if you select this option and a node, and the API using this service does not require a proxy, the request will not be sent through a proxy because one is not defined for that node. If another node requires a proxy, the request is sent through the proxy.

Understanding the Services List Page

The Services page displays all services created in the Management Portal.

Entries for services display the following information:

- The name of the service.
- The status: either **Active** or **Inactive**.
- The service type: **HTTP**,**REST**, or **SOAP**.
- The date and time the service was created and which user created it.

Click a service to view its details page.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.
- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even within words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

  > **Note:**
  >
  > If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Creating a Service

Create an entry for a service you want to manage in the Oracle API Platform Cloud Service - Classic Management Portal.

To create a service:

1. From the Services List page, click **Create Service**.
2. In the **Name** field, enter the name of the service. Note that the name of the service plus the version should be unique.
3. (Optional) In the **Version** field, enter a version number.
4. (Optional) In the **Description** field, enter a brief description of the service.
5. Select the desired type of service from the **Service Type** list.
6. If you selected the HTTP or REST type, enter the endpoint URL and an optional endpoint name.
7. If you selected the SOAP type, do the following:
   a. Click the **WSDL File** button to upload a WSDL file or zip file containing a WSDL file.

      > **Note:**
      >
      > If you upload a zip file, it should only contain XSD and WSDL files.

   b. Click the **Choose File** button, select the file, and click **Open**.
   c. Click **OK** to parse the file.
   d. If there is more than one WSDL file, select a main one from the list of files.
   e. Select the desired port or binding from the **Binding** list.

      The **Endpoint Name** and **Endpoint URL** fields are populated automatically when a port is selected from the **Binding** list. If a binding is selected, **Endpoint Name** is optional, and you must enter an **Endpoint URL**.

      > **Note:**
      >
      > You can create a SOAP service without importing a WSDL file; you just need to specify the endpoint URL. If you do not import a WSDL file, you will not be able to choose the service when creating a REST to SOAP policy.

8. (Optional) Select the **Use Gateway Node Proxy** option if the gateway node requires a proxy to call the service.
9. Click **Create**.

The service is displayed on the Services list page.

Viewing Service Details

You can view the details of a service in a side panel available from any of the tabs.

The side panel displays the following details:

- The name of the service.
- The description of the service.
- The status: either **Active** or **Inactive**.
- The service type: **HTTP**, **REST**, or **SOAP**.
- The most recent date and time that changes were saved for the application and name of the user who saved them.

To view service details:

1. On the Services List page, select the service for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the service.

Editing Service Details

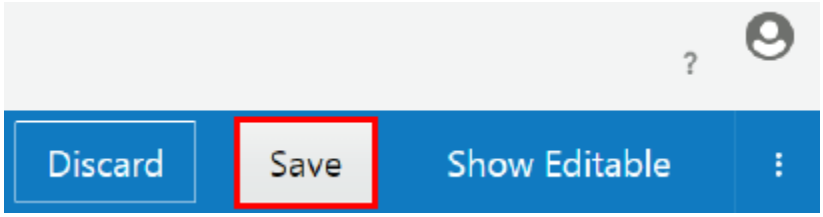
After you create a service, you can then edit its details, including changing the service URL.

To edit service details:

1. From the Services List page, click the service you want to edit.
2. Edit the endpoint name or URL as necessary.
3. (Optional) Click the **Select Account** button and select a select a service account from the list to provide the credentials for the service URL.

   See What Is a Service Account? for more information.
4. Click the drawer icon to display the side panel.

5. Edit the name of the service, the version, or the description in the side panel.
6. Click **Save**.



Deleting a Service

You can delete a service at any time.

To delete a service :

1. On the Services List page, click the service you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes** to confirm.

Managing Service Grants

Service grants allow you to issue fine-grained permissions to users or groups for each service.

Topics

- Understanding Service Grants
- Issuing Service Grants

Understanding Service Grants

Service grants are issues per Service.

Users and groups issued grants for a specific Service have the privileges to perform the associated actions on that Service. See Issuing Service Grants to issue Service grants.

Grant Name	Description	Can be Issued To	Associated Actions
Manage Service	People issued this grant are allowed to view, modify and delete this service.	Service Managers	ServiceEditAll ServiceModifyState ServiceViewAllDetails ServiceViewHistory ServiceDelete ServiceReference ServiceGrantManageService ServiceGrantViewAllDetails ServiceGrantReferenceService
View All Details	People issued this grant are allowed to see all details about this service.	API Managers, Gateway Managers, Service Managers	ServiceViewAllDetails ServiceViewHistory

Grant Name	Description	Can be Issued To	Associated Actions
Reference Service	Users issued this grant are allowed to reference this service (add it to policies).	API Managers, Service Managers	ServiceViewAllDetails ServiceReference

Issuing Service Grants

Grants allow you to control access to services.

To issue grants:

1. From the Services List page, click the service to which you want to issue grants.
2. Click the  (Grants) tab.
3. Click the tab of the grant type you want to issue.
4. Click the **Add Grantee** button.
5. Select the user or group from the **Add to Grant** list. You can select multiple users.
6. Click **Add**.

Understanding the Relationship Between APIs, Services, and Service Accounts

Service accounts and services are resources that you can manage and use in policies for APIs.

A service account defines the security credentials required to invoke a backend service. A service account can either define Basic Auth or OAuth credentials.

A service is used to represent a backend service. It defines the properties required to invoke a backend service. The main required property of the service is the URL at which a backend service can be invoked. A service can also reference a service account to configure the credentials required to invoke a backend service.

An API references services and service accounts through the policies defining the API. An API policy making outbound calls, such as Service Request and Service Callout, can configure the backend service inline by specifying the URL in the policy itself, or the policy can reference the service resource representing the backend service. The policy can also be configured to reference a service account to configure or override credentials information.

Services and service accounts make it easier to manage changes to the services or the required credentials. Update them in one place and all the policies that reference them update to the new values.

6

Managing Plans

A plan is an abstraction between applications (the clients consuming APIs) and APIs to allow fine-grained access entitlements to all APIs that are part of a plan.

Topics:

What is a plan?

Plans are used to group and entitle access for client applications to a set of APIs, enforcing some quality of service or access control criteria.

A plan can allow access to one or multiple APIs. You can also create multiple plans that have access to the same API.

There are three basic use cases for plans:

- Monetization — Levels of cost. For example, you create gold, silver, and bronze plans. The gold plan, at a higher cost, has unlimited access to the API. The silver plan, at a lower cost, has a rate limit, while the bronze level, at the lowest cost, has a smaller . For example, you create a real estate API which can be accessed through bronze, silver, and gold plans. At the bronze level, you can only view 10 listings a day. At the silver level, you can view 50 listings a day. At the gold level, you can view unlimited listings.
- Corporate — Different bundles of APIs. For example, one plan gives employees access to all health care APIs, while another plan gives employees access to an API that lists the company's products.

- Access control — Levels of access. For example, one plan gives internal developers full access to APIs. Another plan gives partners special access to beta APIs for testing. A third plan gives the public limited access to active APIs only.

You can also combine these use cases. In the corporate example, you could have bundles of health care APIs at gold, silver, and bronze cost levels.

Understanding the Plans List Page

The Plans List page displays all plans created in the Management Portal.

Entries for plans display the following information:

- The name and version of the plan.
- The state of the plan: **Active** or **Inactive**.
- The description of the plan.
- The date and time the plan was created and which user created it.
- The number of application subscriptions.
- The number of API entitlements.

> **Note:**
>
> The information you see on this page, and the tabs for a plan, depends on the grants that you have. For example, if you are an API Manager with the View Details Grant, you will only be able to view the Settings, Entitlements, and Publication tabs.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.
- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even within words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

> **Note:**
>
> If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Creating a Plan

Create an entry for a plan you want to manage in the Oracle API Platform Cloud Service - Classic Management Portal.

> **Note:**
>
> Plan names and version numbers do not have to be unique. However, before the plan is published to the Developer Portal, you must create a unique name for the URL in the Developer Portal. Creating multiple plans with the same name is not advisable, however.

To create a plan:

1. From the Plans List page, click **Create Plan**.
2. In the **Plan Name** field, enter the name of the plan.
3. (Optional) In the **Version** field, enter the version of the plan.
4. (Optional) In the **Description** field, enter a brief description of the plan.
5. Click **Create**.

The plan is displayed on the Plans page.
See the following topics to complete the plan management lifecycle.

- Implementing Plans to implement your plan.
- Managing Plan Entitlements to manage the entitlements in your plan.
- Managing Plan Subscriptions to manage the subscriptions to your plan.
- Managing Plan Grants to manage grants to your plan.

Uploading a Plan Icon

You can upload an icon to visually represent a plan in the Management Portal. The icon you upload also represents the plan on the Developer Portal if the plan is published.

For best results, the image you upload should be 60 pixels by 60 pixels. images with other dimensions may be distorted in the Management and Developer Portals.

PNG and JPEG (`.jpg` and `.jpeg`) image formats are supported.

To upload an icon for a plan:

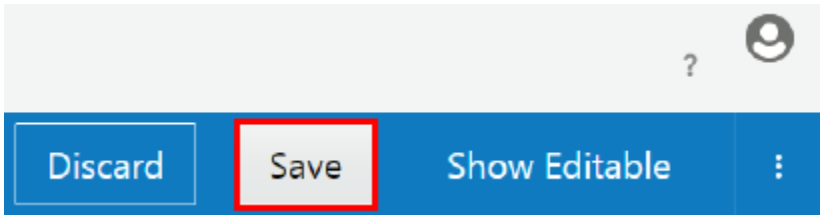
1. On the Plans List page, select a plan.
2. Click the drawer icon to display the side panel.

3. Click the icon to the left of the name of the plan in the side panel.

   The icon dialog appears. The **Custom** tab is selected by default.
4. Choose one of the following options:
   a. To upload a plan icon, click **Choose File**. Select the image you want to use as the API icon, and then click **Open**. Click **OK** to close the dialog.

      > **Tip:**
      >
      > You can also drag and drop an image onto the **Drop File Here** area on the **Custom** tab.

   b. To revert to the default icon, click the **Default** tab, and then click **OK** to close the dialog.
5. Click **Save**.



The plan icon is updated. If you want this icon to represent the plan in the Developer Portal, (re)publish the plan. See Publishing Plans.

Implementing Plans

After you create a plan, you implement it by setting rate limits and selecting gateways.

Topics:

- Setting a Plan Rate Limit
- Setting Plan Gateways

Setting a Plan Rate Limit

Rate limits are a way of giving applications more requests at higher cost levels.

Rate limits apply across all entitlements. For example, you have a plan with entitlements to three different APIs, and then set a rate limit of 1000 requests per minute. This means that requests to all three APIs combined cannot exceed 1000 per minute.

You can set multiple rate limit conditions. In this case, the most restrictive condition is the limiting factor. For example, you set two rate limit conditions, one for 1000 requests per second and another for 10000 requests per minute. The plan allows the full 10000 requests per minute, but if more than 1000 requests occur in any given second, the excess requests are rejected for that second.

To set a rate limit for a plan:

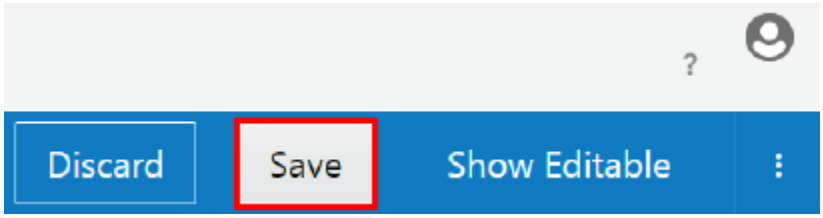
1. On the Plans List page, click the plan for which you want to set a rate limit.

   The Settings page appears.

2. The default rate limit for a plan is **Unlimited** Click **Limited** to set a specific rate limit.
3. Click to enter number of requests.
4. Click the list box and select the time interval. The options are **Second**, **Minute**, **Hour**, and **Day**.

   Click the **Add Condition** (+) icon to add another condition.

5. Click **Save**.



Setting Plan Gateways

You can select one or multiple gateways through which the plan will allow the APIs to be invoked.

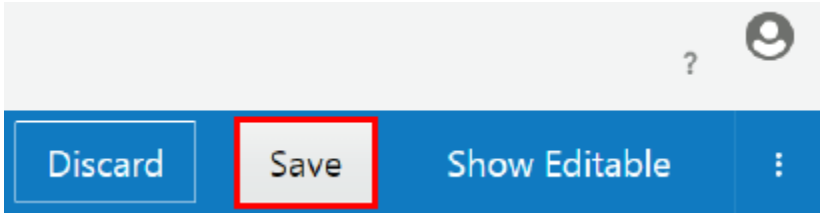
To set gateways for a plan:

1. On the Plans List page, click the plan for which you want to set gateways.

   The Settings page appears.

2. In the **Gateways** section, do one of the following:
   - Click the **All** option to allow the plan to invoke APIs through all gateways to which an API is deployed.
   - Click the **Specific** option to select specific gateways through which the plan can invoke APIs. Click in the field below the option to display a list and select a gateway. To select additional gateways, click in the field again to display the list and choose a gateway.

3. Click **Save**.



Managing Plan Entitlements

An entitlement is the relationship between an API and a plan, describing that a plan entitles a client application to invoke an API, and under what conditions.

Topics:

Understanding Plan Entitlements

An entitlement is the relationship between a plan and an API that defines how a client application can access the API.

There is a many-to-many relationship between plans and APIs. A given plan can have entitlements to multiple APIs; for example, to group related APIs. A given API can be entitled by multiple plans; for example, to provide different quality of service criteria. Note that two entitlements of the same plan cannot point to the same entire API, or to the same action in an API.

You must have a Plan Manager role to manage plan entitlements.

Viewing Plan Entitlement Details

You can view the entitlements in a plan and details such as whether the entitlement is active or published.

To view the details of an entitlement:

1. On the Plans List page, click the plan whose entitlements you want to view.
2. Click the  (Entitlements) tab.
3. To filter the list for active or inactive entitlements, click the **Active** or **Inactive** tab.
4. To filter the list for published or unpublished entitlements, click the **Published** or **Unpublished** tab.
5. To see all entitlements, click the **All** tab.
6. To view more details about an entitlement, click its name or the **Expand** icon to the right.

   In addition to the API name, you can view the description, the user who added the entitlement and when it was added, the rate limit or other constraints that are set.

Adding an API Entitlement to a Plan

A plan can have many entitlements.

To add an entitlement to a plan:

1. On the Plans List page, click the plan to which you want to add an entitlement.
2. Click the  (Entitlements) tab.
3. Click **Add Entitlement**.
4. Select the API or APIs to you want to add to the plan.
5. Click **Add**.

Publishing and Unpublishing an Entitlement in a Plan

Publishing an entitlement makes it availalble to application developers in the Developer Portal.

To publish or unpublish an entitlement in a plan:

1. On the Plans List page, click the plan for which you want to publish or unpublish an entitlement.
2. Click the  (Entitlements) tab.
3. Click the entitlement that you want to publish or unpublish.
4. Click the **Publish** or **Unpublish** button.
5. Click **Yes**.

Activating and Deactivating an Entitlement in a Plan

A plan must be active for it to be available for use.

To activate or deactivate an entitlement in a plan:

1. On the Plans List page, click the plan for which you want to publish or unpublish an entitlement.
2. Click the  (Entitlements) tab.
3. Click the entitlement that you want to publish or unpublish.
4. Click the **Activate** or **Deactivate** button.
5. Click **Yes**.

Removing an Entitlement from a Plan

Removing an entitlement removes access to all the APIs in the entitlement.

To remove an entitlement in a plan:

1. On the Plans List page, click the plan for which you want to remove an entitlement.

2. Click the  (Entitlements) tab.
3. Click the entitlement that you want to remove.
4. Click the **Remove** button.
5. Click **Yes**.

Managing Plan Subscriptions

Applications subscribe to plans to be able to send requests to the APIs in the plan.

Topics:

Understanding Plan Subscriptions

Subscriptions represent the relationship between an application and a plan.

To invoke an API that is entitled to plans, an application must be subscribed to a plan entitling that API. An application cannot subscribe to different plans that have entitlements to the same API.

When an application is subscribed to a plan with multiple API entitlements, it has access to all the APIs in the plan.

Viewing Plan Subscriptions

To view the subscriptions to a plan:

1. From the Plans List page, click the plan for which you want to view subscriptions.
2. Click the  (Subscriptions) tab.

   The Subscriptions page appears. The applications with subscriptions to the plan are displayed.

From this page you can subscribe an application to a plan. You can also approve, reject, suspend, or reactivate application subscriptions to plans.

Subscribing a Plan to an Application

You can create a subscription from an application to a plan from the Plans List page.

You cannot subscribe an application to a plan if the application is already subscribed to a different plan that provides entitlements to the same API or APIs.

To subscribe an application to a plan:

1. On the Plans List page, click the plan to which you want to subscribe an application.
2. Click the  (Subscriptions) tab.
3. Click **Subscribe Application**.
4. Select the application(s) you want to subscribe to the plan.
5. Select an initial subscription state:
   - **Subscribed**: The application is subscribed to the plan.
   - **Requesting Subscription**: The application is subscribed, but the subscription must be approved by an API Manager. See Approving or Rejecting Application Subscriptions to approve subscription requests.
   - **Suspended**: The application is subscribed, but it is in a suspended state. See Resuming a Suspended Application Subscription to activate suspended subscriptions.
6. Click **Subscribe**.

The application is subscribed to the plan.

Approving or Rejecting Plan Subscriptions

A Plan Manager can approve or reject a developer's request to subscribe their application to a plan to get access to APIs entitled by the plan.

Approving the subscription allows an application to send requests to APIs that are entitled by the plan. Until the subscription is approved, or if the subscription is rejected, requests to these APIs are rejected.

To approve or reject an application's subscription to a plan:

1. From the Plans List page, click the application for which a developer has requested for subscription to a plan.
2. Click the  (Subscriptions) tab.
3. Click the **Requesting** tab.
4. Click the name of the application requesting a subscription.
5. Click **Approve** to approve the subscription, **Reject** to reject the subscription, or **Dismiss** to dismiss the subscription.
6. Optionally specify the reason for your action, and then click **Yes** to approve or **No** to reject the subscription.

Approved subscriptions appear on the **Subscribed** tab, while rejected subscriptions appear on the **Rejected** tab. Dismissed subscriptions are deleted.

Suspending Plan Subscriptions

You can temporarily suspend an application's subscription to a plan. While the subscription is suspended, the requests from the application to the entitled APIs in the plan are rejected.

To suspend an application's subscription to a plan:

1. On the Plans List page, click the plan for which you want to suspend an applications's subscription.
2. Click the  (Subscriptions) tab.
3. Click the **Subscribed** tab on the Subscriptions page.
4. Click the application for which you want to suspend the subscription.
5. Click **Suspend**.
6. Optionally specify the reason for suspending the subscription and then click **Yes**.

The application is no longer subscribed to the plan. The plan appears on the **Suspended** tab.
See Resuming a Suspended Plan Subscription to reactivate the subscription.

Resuming a Suspended Plan Subscription

You can resume an application's suspended subscription to a plan.

To resume an application's subscription to a plan:

1. On the Plans List page, click the plan in which you want to resume an application's subscription.
2. Click the  (Subscriptions) tab.
3. Click the **Suspended** tab.
4. Click the application for which you want to resume the subscription.
5. Click **Resume**.
6. Optionally, specify the reason for resuming the subscription and then click **Yes**.

The subscription of the application to the plan is resumed. The application appears on the **Subscribed** tab.

Unsuscribing a Plan

You can unsubscribe an application from a plan to remove its access to the APIs entitled by the plan.

When an application unsubscribes from a plan, Application Developers can no longer see the analytics data for the application in the plan in the API Platform Cloud Service Developer Portal.

To unsubscribe an application from a plan:

1. On the Plans List page, click the plan from which you want to unsubscribe an application.
2. Click the  (Subscriptions) tab.
3. Click the **Subscribed** tab. If the subscription is in the suspended state, click the **Suspended** tab.
4. Click the application you want to unsubscribe from the plan.
5. Click **Unsubscribe**.
6. Click **Yes** in the banner to confirm.

The application is unsubscribed from the plan.

Publishing Plans

Publish a plan to the Developer Portal when you want application developers to discover and consume it.

You can only publish a plan if the portal name has been configured.

Publishing a plan does not automatically publish the APIs entitled in the plan.

To publish a plan to the Developer Portal:

1. On the Plans List page, click the plan you want to publish.
2. Click the  (Publication) tab.
3. Click in the **Developer Portal Name** field and enter a name.
4. In the Publication Summary Information, select the items that you want to be displayed on the plan's summary in the Developer Portal:
   - **Show plan icon**
   - **Show description**
   - **Show rate limit**, with a choice of the rate limit as configured or specific text that you enter
5. (Optional) Click the **Mark as recommended plan** for the plan to be indicated as such in the Developer Portal.
6. Click **Publish to Portal**.

Managing Plan Grants

Plan grants allow you to issue fine-grained permissions to users or groups for specific plans.

Topics:

- Understanding Plan Grants
- Issuing Plan Grants

Understanding Plan Grants

Plan grants are issued per plan.

Users and groups issued grants for a specific plan have the privileges to perform the associated actions on that plan. See Issuing Plan Grantsto issue plan grants.

Grant Name	Description	Can be Issued to	Associated Actions
Manage the plan	Users issued this grant are allowed to modify the definition of and issue users grants for this plan.	Plan Managers	PlanEditAll PlanEditPublic PlanDelete PlanModifyPublishState PlanModifyState PlanViewAllDetails PlanViewPublicDetails PlanViewHistory PlanRequestSubscribeApplication PlanSubscribeApplication PlanApproveSubscription PlanEntitleAPI PlanGrantViewAllDetails PlanGrantViewPublicDetails PlanGrantManagePlan PlanGrantRequestSubscribeApplication PlanGrantSubscribeApplication PlanGrantEntitleAPI
View all details	Users issued this grant are allowed to view all details of this plan in the Management Portal.	API Managers, Gateway Managers, Plan Managers	PlanViewAllDetails PlanViewPublicDetails PlanViewHistory
View public details	Users issued this grant are allowed to see the public details of this plan in the Developer Portal.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails
Subscribe	Users issued this grant are allowed to subscribe applications to this plan.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails PlanSubscribeApplication

Grant Name	Description	Can be Issued to	Associated Actions
Request Subscription	Users issued this grant are allowed to request to subscribe applications to this plan.	API Managers, Application Developers, Plan Managers	PlanViewPublicDetails PlanRequestSubscribeApplication
Entitle	Users issued this grant are allowed to entitle APIs to this plan.	API Managers, Plan Managers	PlanViewPublicDetails PlanEntitleAPI

Issuing Plan Grants

Issue plan grants to users or groups to determine what actions assignees can perform with that API. See Understanding Plan Grants. Grants are issued per plan; repeat this task for each plan you want to issue grants for.

You must be issued the Manage Plan grant for a plan to issue grants for it.

1. On the Plans List page, select the plan for which you want to manage grants.
2. Click the  (Grants) tab.
3. Click the tab that corresponds to the grant you want to issue to users or groups:
   - **Manage the plan**: Plan Manager users issued this grant are allowed to modify the definition of and issue users grants for this plan.
   - **View all details**: Plan Manager, API Manager, and Gateway Manager users issued this grant are allowed to view all details of this plan in the Management Portal.
   - **View public details**: Plan Manager, API Manager, and Application Developer users issued this grant are allowed to see the public details of this plan in the Developer Portal.
   - **Subscribe**: Plan Manager, API Manager, and Application Developer users issued this grant are allowed to subscribe applications to this plan.
   - **Request Subscription**: Plan Manager, API Manager, and Application Developer users issued this grant are allowed to request to subscribe applications to this plan.
   - **Entitle**: Plan Manager and API Manager users issued this grant are allowed to entitle APIs to this plan.
4. Click **Add Grantee**.

   The Add Grantee dialog appears.
5. From the Add Grantee dialog, select the user(s) or group(s) to which you want to issue the grant. You can select multiple users and groups.

> **Note:**
>
> You cannot select users or groups that already have this grant; they are greyed out in the Add Grantee dialog.

6. Click **Add**.

The user(s) or group(s) are issued the plan grant you chose.

Viewing Plan Details

You can view the details of a plan in a side panel available from any of the tabs.

The side panel displays the following details:

- The name and version of the plan.
- The description of the plan.
- The state of publication of the plan, Published, Never Published, or Unpublished.
- The state of the plan: **Active** or **Inactive**.
- The most recent date and time that changes were saved for the API, and name of the user who saved them.

To view plan details:

1. On the Plans List page, select the plan for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the plan.

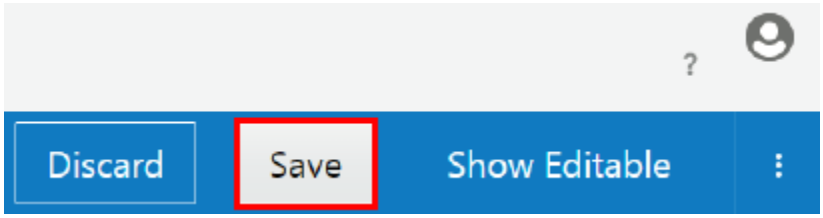
Editing the Plan Description

You can edit the plan description at any time.

To edit the description of a plan:

1. On the Plans List page, click the plan for which you want to edit the description.
2. Click the drawer icon to display the side panel.

3. Edit the description as desired.
4. Click **Save**.



Changing the State of a Plan

A plan can have two states, active and inactive. If a plan is inactive, all requests to APIs made through the plan are rejected.

To change the state of a plan:

1. On the Plans List page, click the plan for which you want to change the state.
2. Click the drawer icon to display the side panel.

3. From the list, select **Active** or **Inactive**.
4. (Optional) Enter comments to describe why the plan is active or inactive.
5. Click **Yes**.

Deleting a Plan

Administrators and Plan Managers can delete plans in the API Platform Cloud Service Management Portal.

You cannot delete a plan if it has entitlements or subscriptions.

To delete a plan:

1. On the Plans List page, click the plan you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes**.

7

Managing Applications

Applications represent the applications API consumers use to send requests to your APIs. Consumers register applications to APIs they use.

Topics

Understanding the Applications List Page

The Applications List page displays all applications created with the Management and Developer portals.

Entries for applications display the following information:

- The name of the application.
- The description of the application.
- The date and time the application was created and which user created it.
- The number of plans to which the application is subscribed.

If you have a long list of items on the page, you can search or sort the list to find the item you want.

- **Sort**: Use the **Sort** list to display the newest items first or display them in alphabetical order.
- **Search**: Use the **Search By Name** field to do a simple search by entering the name of the item you want to find and pressing `Enter`. The search finds the text that you entered anywhere within the name, even within words. For example, if you enter `product`, it will find `production` as well.
- **Advanced Search**: Use the **Advanced** link to create an advanced search query. The link displays a list of fields you can search which are appropriate for the page, such as **Created By**, **Description**, or **Version**. Enter text in the fields to search and click **Apply** to apply all the conditions.
- **Saving a Search**: Once you have performed a search, the conditions you used for the search appear at the top of the list, along with **Save** and **Clear** links. To save the search, click the **Save** link and enter a name for the search. You can also

choose to use it as the default search for the page. To use a saved search, click the list arrow next to the **Search By Name** field and select the search you want to apply.

> **Note:**
>
> If you set a search as a default for a page, the results of that default search appear when you navigate to that page. To view all items, you must clear the search.

- **Editing a Search**: To edit the conditions that a search uses, apply the search, and then add or delete conditions as desired. Save the search with the same name.

Creating an Application

You create applications in the Management Portal. After an application is created, people issued the proper grants can register APIs to, view the details of, or issue grants for the application. See Managing Application Grants.

To create an application:

1. On the Applications List page, click **Create Application**.
2. Complete these fields:
   a. In the **Application Name** field., enter an application name. You can include special characters in the application name.
   b. (Optional) In the **Description** field, enter an application description.
   c. (Optional) From the **Application Types** list, select the type(s) of application you are creating.
3. Click **Create**.

The application is created.

Reissuing an Application Key

You can reissue a key for an application in case it has been compromised, Application keys are established at the application level. If you reissue an application's key, the old key is invalidated. This affects all APIs (that have the key validation policy applied) to which an application is registered. Every request to these APIs must use the new key to succeed. Requests using the old key are rejected. APIs without the key validation policy are not affected as these do not require a valid application key to pass requests.

To reissue an application key:

1. On the Applications List page, select the application for which you want to reissue an application key.
2. Click **Reissue Key**.
3. Click **Yes**.

A new application key is issued.

Managing Application Subscriptions to Plans

Subscriptions to plans allow you to determine to which APIs your applications are allowed to send requests.

Topics

Understanding Application Subscriptions

Subscriptions represent the relationship between an application and a plan. To invoke an API that is entitled to plans, a client must create an application and subscribe it to a plan entitling that API. Application developers use the Developer Portal to subscribe to plans; API managers use the Developer Portal and the Management Portal.
Note that an application cannot subscribe to different plans that have entitlements to the same API.

Analytics use subscriptions to filter data by application:

- API and gateway analytics charts can be filtered to display data from specific applications. Requests from unsubscribed applications are collected as requests from unknown applications.
- Developer Portal analytics charts are always filtered by application. Application developers must subscribe an application to a plan to view data for their requests to an entitled API.

> **Note:**
>
> To view application analytics data in the Developer Portal for requests to an entitled API, the plan, entitlements and API must be published in addition to subscribing an application to a plan.

Viewing Application Subscriptions

You can view the plans to which an application is subscribed.

To view application subscriptions to plans:

1. From the Applications List page, click the application for which you want to view subscriptions.

2. Click the  (Subscriptions) tab.

   The Subscriptions page appears. The plans to which the application has subscribed are displayed.

From this page you can subscribe an application to a plan. You can also approve, reject, suspend, or resume application subscriptions to plans.

Subscribing an Application to a Plan

You can subscribe an application to a plan to enable the application to access the APIs entitled by the plan.

A plan can provide access to several APIs, and an application can subscribe to multiple plans to get access to different APIs. However, an application cannot subscribe to different plans that provide entitlement to the same API. For example, suppose Plan A provides access to API1, API2, and API3, whereas Plan B provides access to API1 and API4. Suppose your application has subscribed to Plan A to get access to API1. Now your application requires access to API4 as well. To access API4, your application cannot subscribe to Plan B because Plan B has a common entitlement with Plan A for API1. Your application needs to unsubscribe to Plan A and then subscribe to Plan B to get access to API1 and API4.

To subscribe an application to a plan:

1. On the Applications List page, click the application that you want to subscribe to a plan.
2. Click the  (Subscriptions) tab.
3. Click **Subscribe to Plan**.
4. Select the plan(s) to which you want to subscribe.
5. Select an initial subscription state:
   - **Subscribed**: The application is subscribed to the plan.
   - **Requesting**: The application is subscribed, but the subscription must be approved by an API Manager. See Approving or Rejecting Application Subscriptions to approve subscription requests.
   - **Suspended**: The application is subscribed, but it is in a suspended state. See Resuming a Suspended Application Subscription to resume suspended subscriptions.
6. Click **Subscribe**.

The application is subscribed to a plan.

Approving or Rejecting Application Subscriptions

A Plan Manager can approve or reject a developer's request to subscribe their application to a plan to get access to APIs entitled by the plan.

Approving the subscription allows an application to send requests to APIs that are entitled by the plan. Until the subscription is approved, or if the subscription is rejected, requests to these APIs are rejected.

To approve or reject an application's subscription to a plan:

1. From the Applications List page, click the application for which a developer has requested for subscription to a plan.
2. Click the  (Subscriptions) tab.
3. Click the **Requesting** tab.
4. Click the name of the plan to which subscription is requested.
5. Click **Approve** to approve the subscription, **Reject** to reject the subscription, or **Dismiss** to dismiss the subscription.
6. Optionally specify the reason for your action, and then click **Yes** to approve or **No** to reject the subscription.

Approved subscriptions appear on the **Subscribed** tab, while rejected subscriptions appear on the **Rejected** tab. Dismissed subscriptions are deleted.

Suspending Application Subscriptions

You can temporarily suspend an application's subscription to a plan. While the subscription is suspended, the requests from the application to the entitled APIs in the plan are rejected.

To suspend an application's subscription to a plan:

1. On the Applications List page, click the application for which you want to suspend subscription to a plan.
2. Click the  (Subscriptions) tab.
3. Click the **Subscribed** tab on the Subscriptions page.
4. Click the plan for which you want to suspend the application's subscription.
5. Click **Suspend**.
6. Optionally specify the reason for suspending the subscription and then click **Yes**.

The application is no longer subscribed to the plan. The plan appears on the **Suspended** tab.
See Resuming a Suspended Application Subscription to reactivate the subscription.

Resuming a Suspended Application Subscription

You can resume an application's suspended subscription to a plan.

To resume an application's subscription to a plan:

1. On the Applications List page, click the application for which you want to reactivate a subscription.
2. Click the  (Subscriptions) tab.
3. Click the **Suspended** tab.
4. Click the plan for which you want to resume the application's subscription.

5. Click **Resume**.
6. Optionally, specify the reason for resuming the subscription and then click **Yes**.

The subscription of the application to the plan is resumed. The plan appears on the **Subscribed** tab.

Unsubscribing an Application

You can unsubscribe an application from a plan to remove its access to the APIs entitled by the plan.

When an application unsubscribes from a plan, Application Developers can no longer see the analytics data for the application in the plan in the API Platform Cloud Service Developer Portal.

To unsubscribe an application from a plan:

1. On the Applications List page, click the application that you want to unsubscribe from a plan.
2. Click the  (Subscriptions) tab.
3. Click the **Subscribed** tab. If the subscription to the plan is in the suspended state, click the **Suspended** tab.
4. Click the plan from which you want to unsubscribe the application.
5. Click **Unsubscribe**.
6. Click **Yes** in the banner to confirm.

The application is unsubscribed from the plan.

Managing Application Grants

Application grants allow you to issue fine-grained permissions to users or groups for specific applications.

Topics

Understanding Application Grants

Application grants are issued per application.

Users issued grants for a specific application have the privileges to perform the associated actions on that application. See Issuing Application Grants to issue application grants.

Grant Name	Description	Can be Issued To	Associated Actions
Manage Application	People issued this grant can view, modify and delete this application. API Manager users issued this grant can also issue grants for this application to others.	API Managers, Application Developers, Plan Managers	ApplicationEdit ApplicationDelete ApplicationView ApplicationGrantManageApplication
View All Details	People issued this grant can see all details about this application in the Developer Portal.	API Managers, Application Developers, Plan Managers	ApplicationViewAllDetails

Issuing Application Grants

You can issue grants to users that allow them to view application details or manage applications in the Management Portal or the Developer Portal.

Grants are issued per application; repeat this task for each application you want to issue grants for.

You must be issued the Manage Application grant to issue grants for it.

1. On the Applications List page, select the application for which you want to manage grants.
2. Click the  (Grants) tab.
3. Click the tab that corresponds to the grant you want to issue to users or groups:
   - **Manage Application**: users issued this grant can modify and delete this application. API Manager users issued this grant can also issue grants for this application to other users.
   - **View all details**: users issued this grant can see all details about this application in the Management Portal, or in the Developer Portal, in the case of Application Developer users.
4. Click **Add Grantee**.

   The Add Grantee dialog appears.
5. From the Add Grantee dialog, select the user(s) or group(s) to which you want to issue the grant. You can select multiple users and groups.

   > **Note:**
   >
   > You cannot select users or groups that already have this grant; they are greyed out in the Add Grantee dialog.
6. Click **Add**.

The user(s) or group(s) are issued the application grant you chose.

Viewing Application Details

You can view the details of an application in a side panel available from any of the tabs.

The side panel displays the following details:

- The name of the application.
- The description of the application.
- The most recent date and time that changes were saved for the application, and name of the user who saved them.

To view application details:

1. On the Applications List page, select the application for which you want to view details.
2. Click the drawer icon to display the side panel.

The side panel opens, displaying the details for the application.

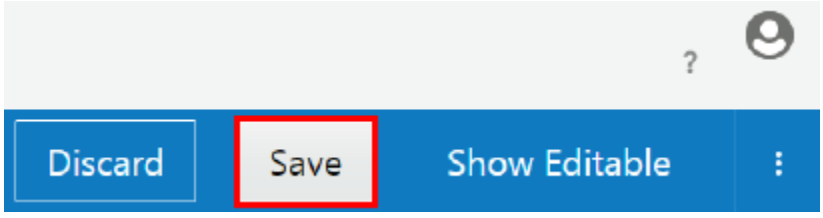
Editing Application Details

You can edit the name and description of an application at any time.

To edit application details:

1. On the Application List page, select the application for which you want to edit details.
2. Click the drawer icon to display the side panel.

3. Edit the name and description as desired.
4. Click **Save**.



Delete an Application

Administrators and API Managers can delete applications in the Management Portal.

You cannot delete an application if it is registered to APIs or you don't have the Manage Application grant for the application. If you cannot delete an application, the Delete button is grayed out. Ensure you unregister it from all APIs and that you have the proper grant before trying again.

To delete an application:

1. On the Applications List page, click the application you want to delete.
2. Click the drawer icon to display the side panel.

3. Click **Delete**.
4. Click **Yes** in the banner to confirm.

The application is deleted.

8
Use Analytics

Analytics charts in the Oracle API Platform Cloud Service - Classic Management Portal show you critical information, like who is using your API, how APIs are being used, and if errors are occurring.

See Using Application Analytics in *Consuming APIs with the Oracle API Platform Cloud Service Developer Portal* for analytics charts in the Oracle API Platform Cloud Service - Classic Developer Portal.

Topics

Viewing API Analytics

You can use analytics to determine how, when, and why your APIs are being used, review how often and why requests are rejected, and monitor data trends.

API Managers must be issued the Manage API or View All Details API grant to view analytics for that API.

To view API analytics:

1. From the APIs List page, click the name of the API that you want to view analytics for.
2. Click the  (Analytics) tab.

   Analytics data for this API appears. The General page appears by default.
3. Click the **General**, **Applications**, or **Errors and Rejections** tabs to view the available charts.

API Analytics Charts Available on the General Page

View the request volume, response time, and other metrics about requests sent to your APIs on the General page.

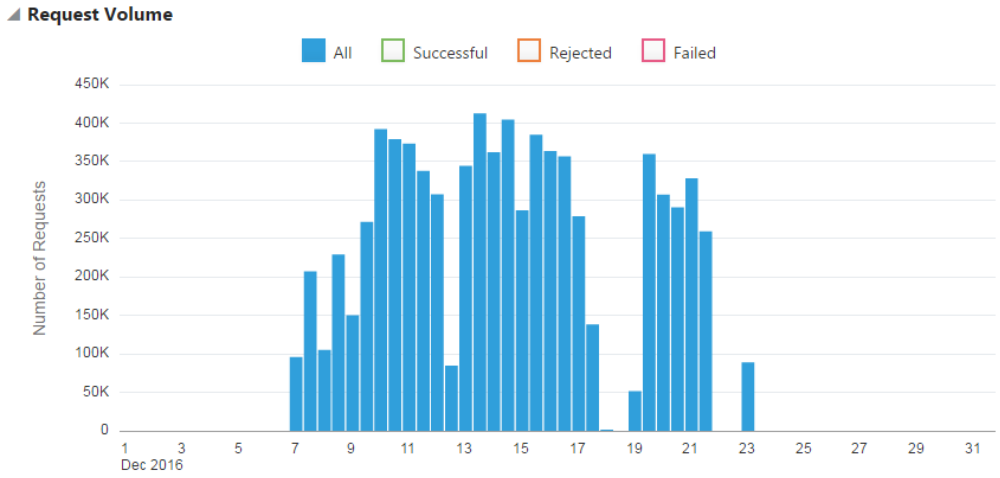
Request Volume Chart

Use the Request Volume chart to view request traffic volume, request trends, and request success or failure rates. The Request Volume chart displays the number of requests sent to an API.

To view Request Volume chart data:

- Select **All** to display all requests.

- Select **Successful** to display successful requests.
- Select **Rejected** to display rejected requests.
- Select **Failed** to display requests that failed.



To view summary information for a specific period, hover your mouse over a vertical bar in the Request Volume chart.

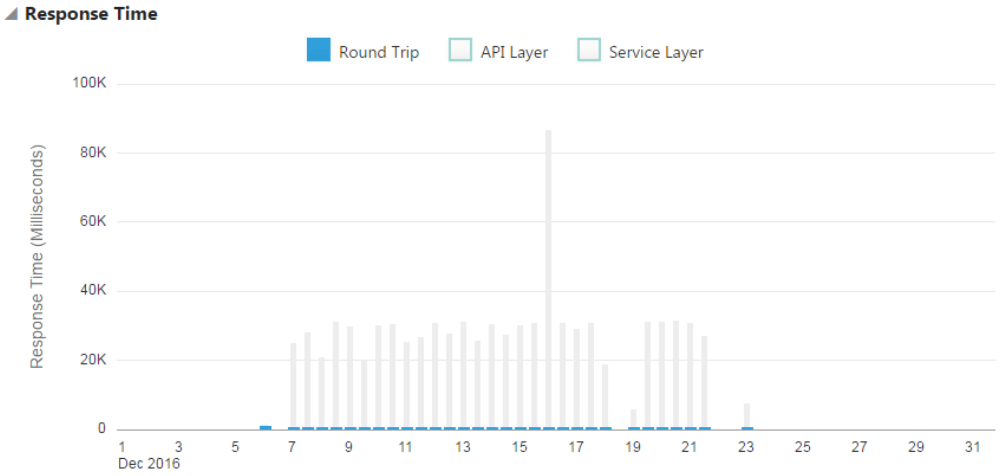
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Response Time Chart

Use the Response Time chart to view the median response times for requests, response time trends, and the time requests spend in the API and service layers. The Response Time chart displays round-trip request and response times (in milliseconds) for the selected API. The shortest and longest response times for the period are represented by vertical bars. A horizontal bar indicates the median round-trip time for the period.

To view Response Time chart data:

- Select **Round Trip** to display the median, the shortest, and the longest round-trip request and response times for the period.
- Select **API Layer** to display the median, the shortest, and the longest time that requests and responses were active in the API layer for the period.
- Select **Service Layer** to display the median, the shortest, and the longest time that requests and responses were active in the service layer for the period.



To view summary information for a specific period, hover your mouse over a vertical or horizontal bars in the Response Time chart.

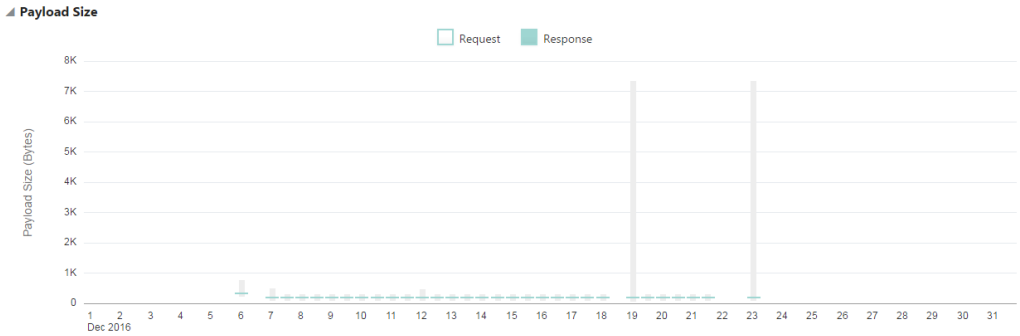
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Payload Size Chart

The Payload Size chart displays the size of the payload sent with each request. The largest and smallest payload sizes for the period are represented by vertical bars. A horizontal bar indicates the median payload size during the period.

To view Payload Size chart data:

- Select **Request** to view request payload sizes for the period.
- Select **Response** to view response payload sizes for the period.



To view summary information for a specific period, hover your mouse over a vertical bar in the Payload Size chart.

By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Requests by Resource Chart

The Requests by Resource chart displays the volume and distribution of requests (as a percentage) to resources for an API.

The Requests by Resource chart displays this data:

- The volume of requests sent to each resource.
- The percentage of requests sent to each resource.
- The volume of rejected requests for each resource.
- The percentage of rejected requests for each resource.
- The volume of errors processed by each resource.
- The percentage of errors processed by each resource.

Requests By Resource

Request Path	Requests		Rejections		Errors	
	Volume	Distribution	Volume	Distribution	Volume	Distribution
/BookStoreAPI/user/bookstore	10	41.7%				
/BookStoreAPI/calloutservices/apics/throwinternalservererror	8	33.3%			8	100.0%
/BookStoreAPI/user/apics/testdelete	2	8.3%				
/BookStoreAPI/user/bookstore/	2	8.3%	2	100.0%		
/BookStoreAPI/user/apics/testput	2	8.3%				

API Analytics Charts Available on the Applications Page

You can see how many requests, per application, have been sent to your APIs on the Applications page of the Analytics page.

Active Applications Chart

The Active Applications chart displays the requests, rejections, and errors resulting from requests to an API. Applications are identified by application keys passed with each request. Requests passed without application keys are collected in an Unknown Applications entry in the chart. If the API is not secured by a key validation policy, all requests, regardless of whether application keys are passed, are collected in the Unknown Applications entry.

The Active Applications chart displays this data:

- The volume of requests sent from each application.
- The percentage of requests sent from each application.
- The volume of rejected requests sent from each application.
- The percentage of rejected requests sent from each application.
- The error volume for requests sent from each application.
- The error percentage for requests sent from each application.

Active Applications

Application	Requests		Rejections		Errors	
	Volume	Distribution	Volume	Distribution	Volume	Distribution
Unknown Application (No Key)	7954117	100.0%	103983	100.0%	7884	99.9%
Unknown Application (ID 371)	8	0.0%	5	0.0%	3	0.0%
testwithouttype	5	0.0%			4	0.1%

API Analytics Charts Available on the Errors and Rejections Page

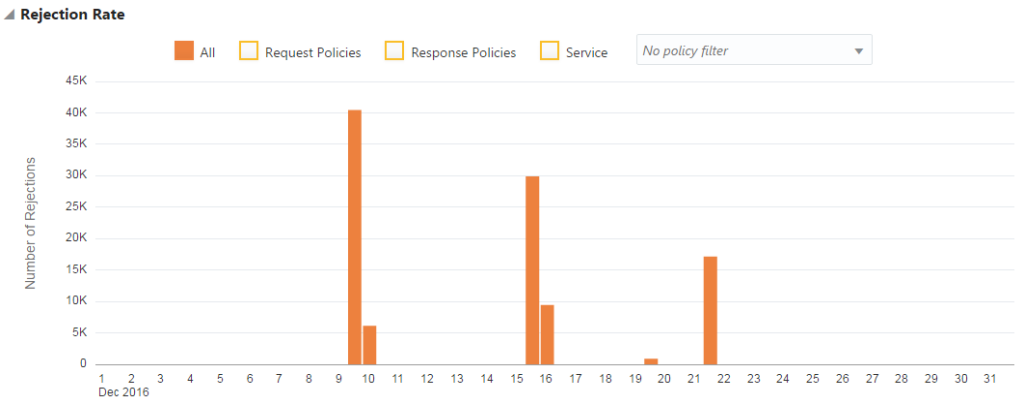
View rejection and error metrics about requests sent to your APIs on the Errors and Rejections page.

Rejection Rate Chart

The Rejection Rate chart displays the number of rejected requests sent to an API.

To view Rejection Rate chart data:

- Select **All** to view all rejections for the period.
- Select **Request Policies** to view requests that were rejected by request policies for the period.
- Select **Response Policies** to view responses that were rejected by response policies for the period.
- Select **Service** to view requests that were rejected by the backend service.
- Select an option in the **no policy filter** list: select a policy type (like **Header Validation**) to view data for all policies of that type, or select the name of a specific policy instance (like **tenant-id validation**, when that is the name of a specific header validation policy) to view data for that specific policy instance.



To view summary information for a specific period, hover your mouse over a vertical bar in the Rejection Rate chart.

By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

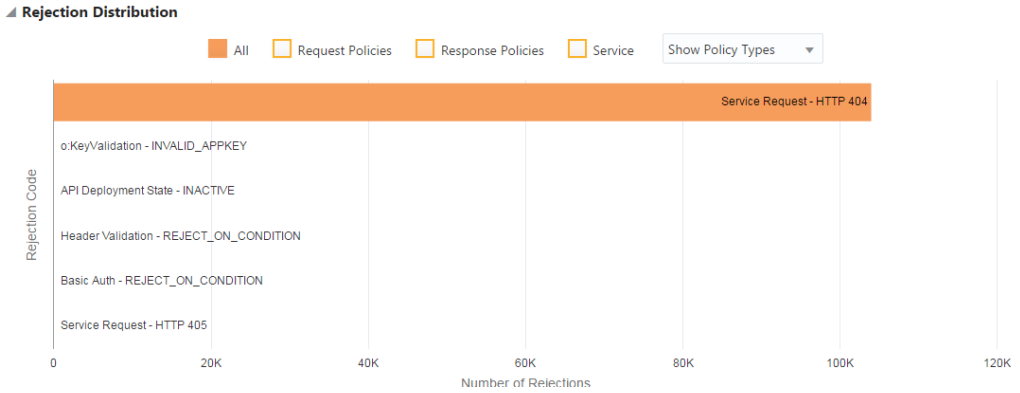
Rejection Distribution Chart

The Rejection Distribution chart displays the number of rejections by specific policies or services.

To view Rejection Distribution chart data:

- Select **All** to view all rejections per policy or service.
- Select **Request Policies** to view the number of rejections per request policy.
- Select **Response Policies** to view the number of requests per response policy.

- Select **Service** to view the number of rejections per service.
- Select **Show Policy Types** to view rejections for each policy type. For example, if you have multiple header validation policies, selecting **Show Policy Types** displays all header validation policy rejections as a single data point.
- Select **Show Policy Instances** to view rejections for each instance of a policy. For example, if you have multiple header validation policies, selecting **Show Policy Instances** displays rejections from each of the header validation policies as separate data points.



To view summary information for a specific policy, hover your mouse over a horizontal bar in the Rejection Distribution chart.

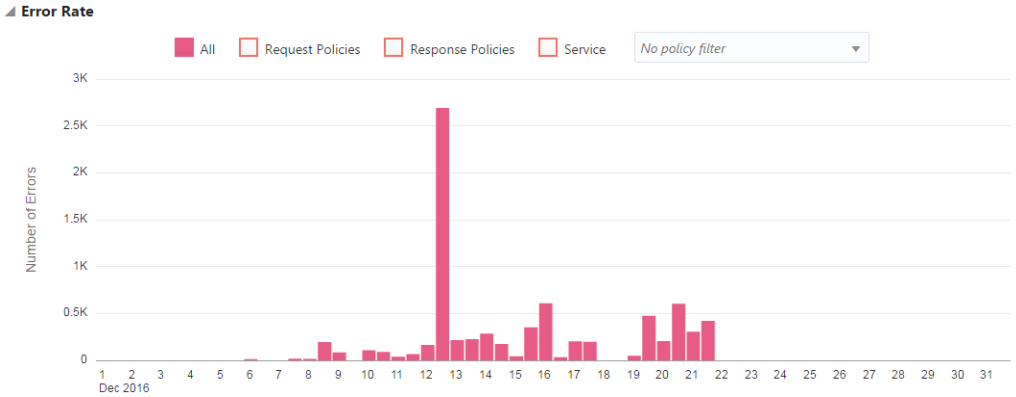
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Error Rate Chart

The Error Rate chart displays the number of errors as a percentage of all errors for the defined period.

To view Error Rate chart data:

- Select **All** to view all errors for the period.
- Select **Request Policies** to view errors caused by request policies for the period.
- Select **Response Policies** to view errors caused by response policies for the period.
- Select **Service** to view errors caused by requests rejected by the backend service.
- Select **Show Policy Types** to view errors for each policy type.
- Select **Show Policy Instances** to view errors for each instance of a policy.



To view summary information for a specific period, hover your mouse over a vertical bar in the Error Rate chart.

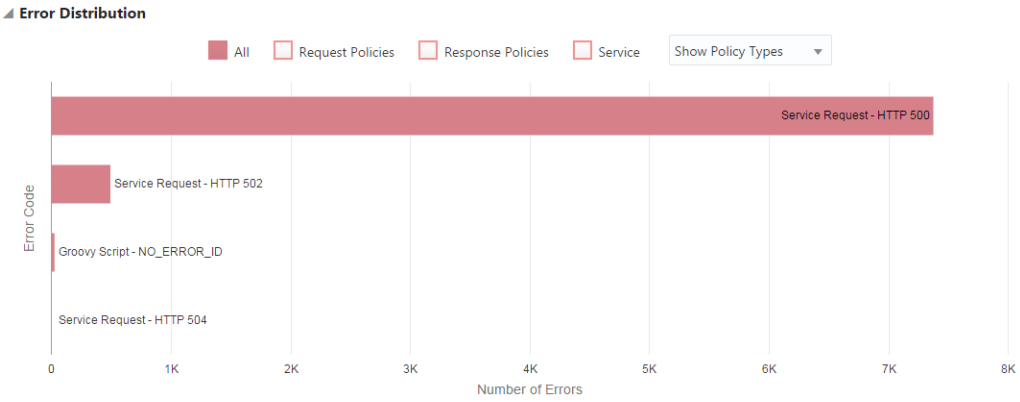
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Error Distribution Chart

The Error Distribution chart displays the number of API request errors for specific policies.

To view Error Distribution chart data:

- Select **All** to view all errors.
- Select **Request Policies** to view errors caused by request policies.
- Select **Response Policies** to view errors caused by response policies.
- Select **Service** to view errors caused by the backend service.
- Select **Show Policy Types** to view errors for each policy type. For example, if you have multiple header validation policies, selecting **Show Policy Types** displays all header validation policy rejections as a single data point.
- Select **Show Policy Instances** to view errors for each instance of a policy. For example, if you have multiple header validation policies, selecting **Show Policy Instances** displays rejections from each of the header validation policies as separate data points.



To view summary information for a specific policy, hover your mouse over an entry in the Error Distribution chart.

Viewing Gateway Analytics

You can use analytics to determine how, when, and why requests are sent to your gateways, review how often and why requests are rejected, and monitor data trends.

Gateway Managers must be issued the Manage Gateway or View All Details gateway grant to view analytics for that gateway.

To view gateway analytics:

1. From the **Gateways List** page, select the gateway that you want to view analytics for.
2. Click the  (Analytics) tab.

   Analytics data for this gateway appears. The General page appears by default.
3. Click the **General**, **Applications**, or **Errors and Rejections** pages to view the available charts.

Gateway Analytics Charts Available on the General Page

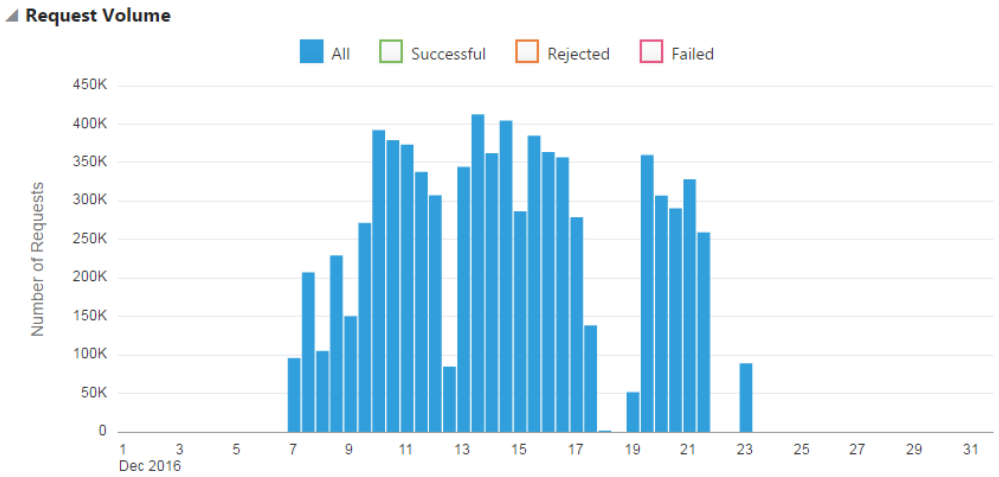
View the request volume, response time, and other metrics about requests sent to your gateways on the General page.

Request Volume Chart

Use the Request Volume chart to view request traffic volume, request trends, and request success or failure rates. The Request Volume chart displays the number of requests sent to an all APIs deployed to a gateway.

To view Request Volume chart data:

- Select **All** to display all requests.
- Select **Successful** to display successful requests.
- Select **Rejected** to display rejected requests.
- Select **Failed** to display requests that failed.



To view summary information for a specific period, hover your mouse over a vertical bar in the Request Volume chart.

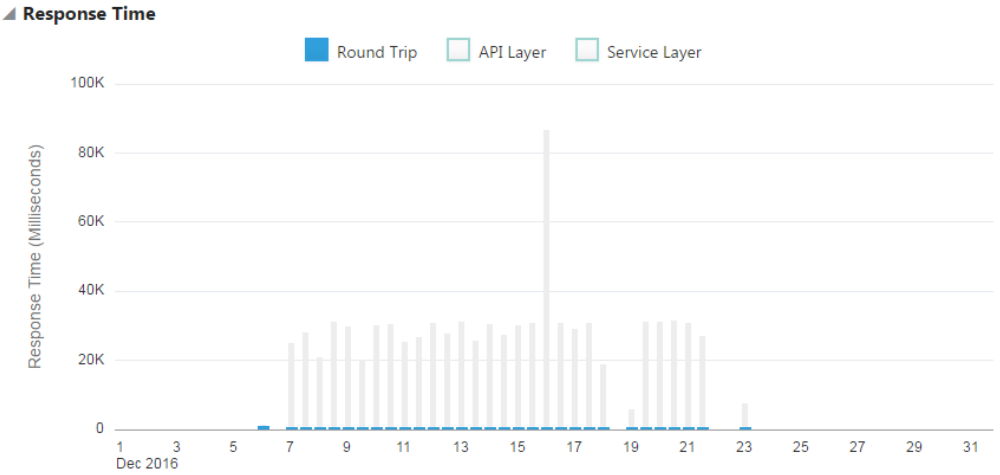
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Response Time Chart

Use the Response Time chart to view the median response times for requests, response time trends, and the time requests spend in the API and service layers. The Response Time chart displays round-trip request and response times (in milliseconds) for the selected gateway. The shortest and longest response times for the period are represented by vertical bars. A horizontal bar indicates the median round-trip time for the period.

To view Response Time chart data:

- Select **Round Trip** to display the median, the shortest, and the longest round-trip request and response times for the period.
- Select **API Layer** to display the median, the shortest, and the longest time that requests and responses were active in the API layer for the period.
- Select **Service Layer** to display the median, the shortest, and the longest time that requests and responses were active in the service layer for the period.



To view summary information for a specific period, hover your mouse over a vertical or horizontal bars in the Response Time chart.

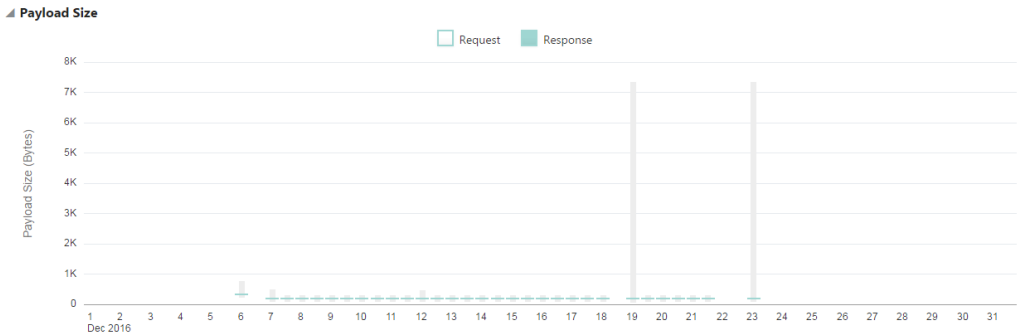
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Payload Size Chart

The Payload Size chart displays the size of the payload sent with each request. The largest and smallest payload sizes for the period are represented by vertical bars. A horizontal bar indicates the median payload size during the period.

To view Payload Size chart data:

- Select **Request** to view request payload sizes for the period.
- Select **Response** to view response payload sizes for the period.



To view summary information for a specific period, hover your mouse over a vertical bar in the Payload Size chart.

By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Requests by API Chart

The Requests by API chart displays requests, rejections, and errors for each API deployed to the gateway.

The Requests by API chart displays this data:

- The volume of requests sent to each API.
- The percentage of requests sent to each API.
- The volume of rejected requests for each API.
- The percentage of rejected requests for each API.
- The error volume for each API.
- The error percentage for each API.

Requests by API

API	Requests		Rejections		Errors	
	Volume	Distribution	Volume	Distribution	Volume	Distribution
weather	1590929	20.0%	103967	100.0%	4237	53.7%
apiswd	1590928	20.0%	8	0.0%	13	0.2%
test	1590783	20.0%	1	0.0%	1156	14.6%
testbug1	1590666	20.0%			1222	15.5%

Requests by Resource Chart

The Requests by Resource chart displays the volume and distribution of requests (as a percentage) to resources for APIs deployed to your gateway.

The Requests by Resource chart displays this data:

- The volume of requests sent to each resource.
- The percentage of requests sent to each resource.
- The volume of rejected requests for each resource.
- The percentage of rejected requests for each resource.
- The volume of errors processed by each resource.
- The percentage of errors processed by each resource.

Requests By Resource

Request Path	Requests		Rejections		Errors	
	Volume	Distribution	Volume	Distribution	Volume	Distribution
/test	13	0.0%			13	0.2%
/customer/1	1590929	20.0%	103967	100.0%	4237	53.7%
/webpost/posts/1	1590764	20.0%			1156	14.6%
/time	1590666	20.0%			1222	15.5%

Gateway Analytics Charts Available on the Applications Page

You can see how many requests, per application, have been sent to your gateways on the Applications page of the Analytics tab.

Active Applications Chart

The Active Applications chart displays the requests, rejections, and errors resulting from requests to all APIs deployed to a gateway. Applications are identified by application keys passed with each request. Requests passed without application keys are collected in an Unknown Applications entry in the chart. If the API is not secured by a key validation policy, all requests, regardless of whether application keys are passed, are collected in the Unknown Applications entry.

The Active Applications chart displays this data:

- The volume of requests sent from each application.
- The percentage of requests sent from each application.
- The volume of rejected requests sent from each application.

- The percentage of rejected requests sent from each application.
- The error volume for requests sent from each application.
- The error percentage for requests sent from each application.

Active Applications

Application	Requests		Rejections		Errors	
	Volume	Distribution	Volume	Distribution	Volume	Distribution
Unknown Application (No Key)	7954117	100.0%	103983	100.0%	7884	99.9%
Unknown Application (ID 371)	8	0.0%	5	0.0%	3	0.0%
testwithouttype	5	0.0%			4	0.1%

Gateway Analytics Charts Available on the Errors and Rejections Page

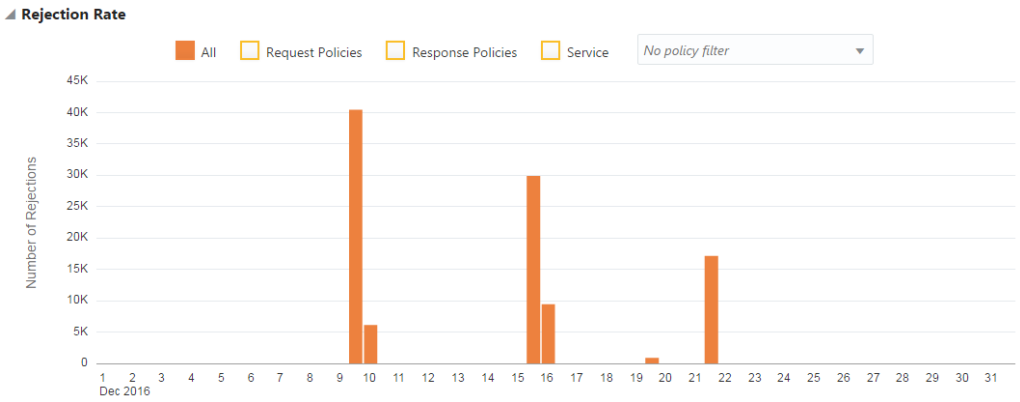
View rejection and error metrics about requests sent to your gateways on the Errors and Rejections page.

Rejection Rate Chart

The Rejection Rate chart displays the number of rejected requests sent to all APIs deployed to a gateway.

To view Rejection Rate chart data:

- Select **All** to view all rejections for the period.
- Select **Request Policies** to view requests that were rejected by request policies for the period.
- Select **Response Policies** to view responses that were rejected by response policies for the period.
- Select **Service** to view requests that were rejected by the backend service.
- Select an option in the **no policy filter** list: select a policy type (like **Header Validation**) to view data for all policies of that type, or select the name of a specific policy instance (like **tenant-id validation**, when that is the name of a specific header validation policy) to view data for that specific policy instance.



To view summary information for a specific period, hover your mouse over a vertical bar in the Rejection Rate chart.

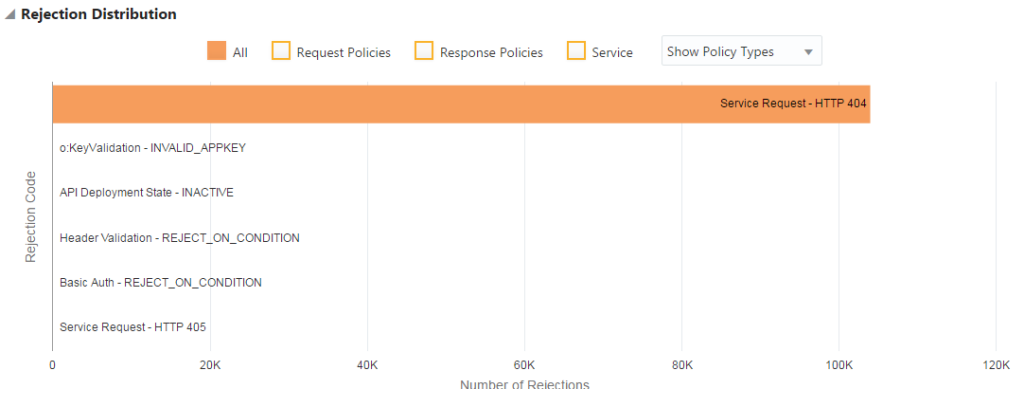
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Rejection Distribution Chart

The Rejection Distribution chart displays the number of rejections by specific policies or services.

To view Rejection Distribution chart data:

- Select **All** to view all rejections per policy or service.
- Select **Request Policies** to view the number of rejections per request policy.
- Select **Response Policies** to view the number of rejections per response policy.
- Select **Service** to view the number of rejections per service.
- Select **Show Policy Types** to view rejections for each policy type. For example, if you have multiple header validation policies, selecting **Show Policy Types** displays all header validation policy rejections as a single data point.
- Select **Show Policy Instances** to view rejections for each instance of a policy. For example, if you have multiple header validation policies, selecting **Show Policy Instances** displays rejections from each of the header validation policies as separate data points.



To view summary information for a specific policy, hover your mouse over a horizontal bar in the Rejection Distribution chart.

By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

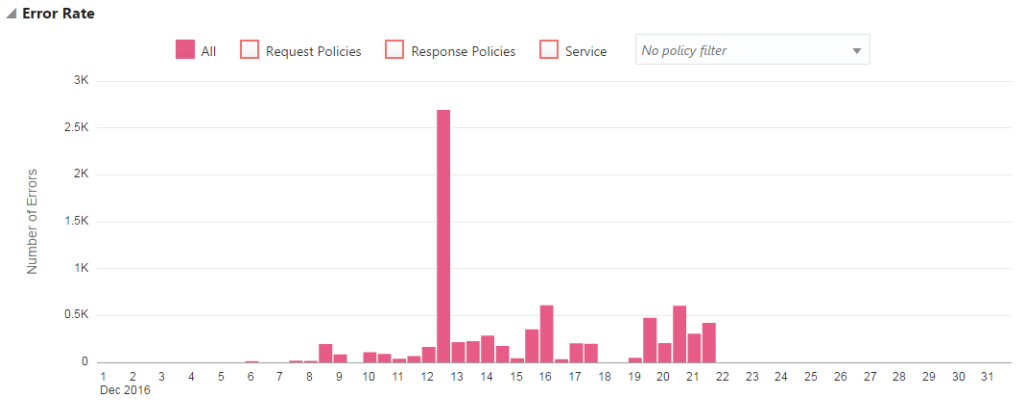
Error Rate Chart

The Error Rate chart displays the number of errors for the defined period.

To view Error Rate chart data:

- Select **All** to view all errors for the period.
- Select **Request Policies** to view errors caused by request policies for the period.

- Select **Response Policies** to view errors caused by response policies for the period.
- Select **Service** to view errors caused by backend service.
- Select an option in the **no policy filter** list: select a policy type to view data for all policies of that type, or select the name of a specific policy instance to view data for that specific policy instance.



To view summary information for a specific period, hover your mouse over a vertical bar in the Error Rate chart.

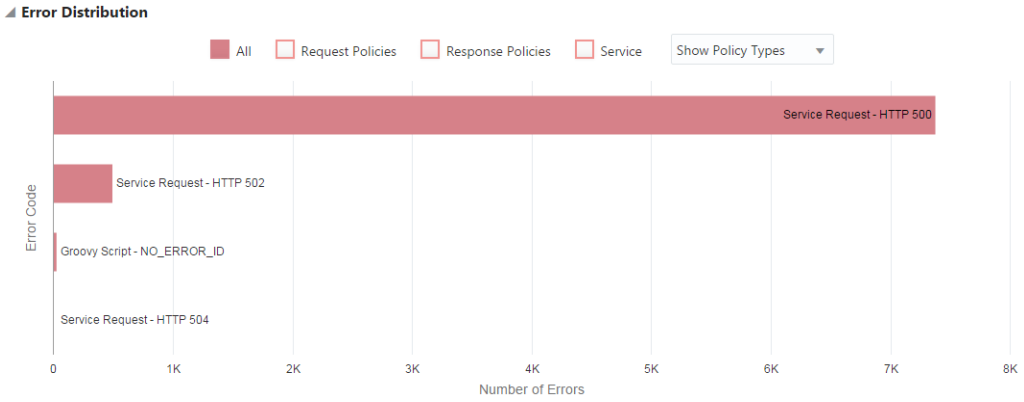
By default, data for the current day is displayed. To display data for a different period, see Working with Analytics Time Controls.

Error Distribution Chart

The Error Distribution chart displays the number of API request errors for specific policies.

To view Error Distribution chart data:

- Select **All** to view all errors.
- Select **Request Policies** to view errors caused by request policies.
- Select **Response Policies** to view errors caused by response policies.
- Select **Service** to view errors caused by the backend service.
- Select **Show Policy Types** to view errors for each policy type. For example, if you have multiple header validation policies, selecting **Show Policy Types** displays all header validation policy rejections as a single data point.
- Select **Show Policy Instances** to view errors for each instance of a policy. For example, if you have multiple header validation policies, selecting **Show Policy Instances** displays rejections from each of the header validation policies as separate data points.



To view summary information for a specific policy, hover your mouse over an entry in the Error Distribution chart.

Working with Analytics Time Controls

Time controls enable you to retrieve analytics data for specific time periods.

By default, data from today (from 12:00AM, expressed in the default time zone set for the platform, to the current time) appears. To change the displayed analytics data time period:

Click **Last 24 Hours** to display data for that period, stretching back 24 hours from the current time. If you want to display data over a different period, you can either:

- Select a pre-defined time period (current hour or week, specific month or year, last 15 minutes, etc.) from the **Other** list, or
- Click the time interval below the **Other** list to manually specify a start and end time and date.

Analytics data for the time period you selected appears.

Filtering Analytics

You can filter the analytics data to display only information for specific gateways, applications, or APIs, depending on your role.

To filter analytics data:

- Navigate to the Analytics page for your API or Gateway. See Viewing API Analytics or Viewing Gateway Analytics.
- **(For API Analytics only)**: In the **Gateways** field, enter gateways on which an API is deployed to display data for only these gateways. If nothing is entered in the Gateways field data for all gateways appears.

> **Note:**
>
> If you are not issued appropriate grants to view gateways the API is deployed to, or if the API is not deployed to a gateway, you are unable to filter by gateway and the text **No Gateways Available** is displayed.

- In the **Applications** field, enter applications registered to an API to display data for only these applications. Requests received without application keys (which is how requests from applications are identified) are also collected; you can view data for all of these requests by selecting **Unknown Applications** from this list. If nothing is entered in the Applications field, data for all applications, including unknown applications, appears.
- **(For gateway analytics only)** In the **APIs** field, enter APIs deployed to a gateway to display data for only these APIs. If nothing is entered into the APIs field, data for all APIs appears.

> **Note:**
>
> If you are not issued appropriate grants to view APIs deployed to the gateway, or if there are no APIs deployed to your gateways, you are unable to filter by API and the text **No APIs Available** is displayed.

9

Frequently Asked Questions for Oracle API Platform Cloud Service - Classic

This is a short list of our most frequently asked questions.

Topics

How is the Oracle Data Model Superior to its Competitors?

These features make the Oracle data model superior to its competitors:

- Deployment — One API can be deployed to multiple gateways. The identity of the API is maintained as a single entity making it easier to manage. The iteration history allows you to quickly determine which API iteration is deployed to a specific gateway.
- Grants — A grant is a relationship between a user and an object. Grants allow you to manage the permissions you assign to users or groups to access specific objects. For example, a user with permissions to issue grants can assign a grant to another user.

- Plans and OAuth — The Oracle data model supports this relationship: Application <> Registration <> Plan <> APIPlan Contract <> API. This relationship allows the application developer to choose the plan they want to use to access an API. An API can be associated with multiple plans, a plan can be associated with multiple APIs, and an application can be registered to multiple plans. You determine where the Oauth Token is used. This flexibility allows you to create custom solutions to meet your specific requirements.

Are API Manager, API Catalog, and API Gateway used with API Platform?

No. Although API Platform has similar capabilities to the other products, it is a unique application. The only shared component is the Oracle Communications Services Gatekeeper (OCSG) server runtime engine.

Does My Service Stop when the Number of Allowed Requests are Exceeded?

No. Requests will continue to be processed and you will be notified that you have exceeded your subscription limit. If you do not purchase additional subscriptions or credits, management service access is discontinued.

Does API Platform Have API Harvesting Capabilities?

No. With API Platform, you model APIs and services to expose the REST APIs needed to create and manage the API objects and services.

Can I Use APIs to Automate or Extend the Capabilities of API Platform?

Yes. You can use APIs to extend the capabilities of API Platform including modifying the history table, localizing the development portal, customizing the development portal, and managing API iterations.

Are Unknown Developer Portal Users Supported?

No. Support for unknown developer portal users is not supported in this release.

Is API Cloning Supported?

Yes. See Cloning an API.

Are SOAP APIs Supported?

SOAP APIs are supported. Gateways handle requests to SOAP services as an HTTP passthrough. You can use any policy with SOAP, but certain policies do not make much sense to use with SOAP.

For example, a method mapping which takes the HTTP resource/verb and maps it to different HTTP resource/verb combination. SOAP is all HTTP POST

The OAuth2 policy is another example. Yes, you can use OAuth2 with SOAP, but the implementation uses a JSON web-token and SOAP is XML. While possible, it would be a bit awkward to mix/match the two.

Interface filtering is yet another because there is only one resource (and verb) combination with SOAP.

You **can** use rate limits and other policies with SOAP and this follows a best practice.

Can Requests be Routed to the Nearest Gateway or to a Different Instance of the Underlying Service?

No. Requests are routed to the gateway before the API is invoked. To route requests to the nearest gateway or a different instance of the underlying service, use gateway based routing to invoke a configured service for a specific gateway. To enable gateway based routing, deploy a single API to send requests to the specified gateway service.

Can I View a History of User Activity or API Iterations?

Yes. Use the REST API to view a history of user activity. Use the `.../apis/{api_id}/iterations/{iteration_id}` resource to view API iteration history.

Does the API Gateway Allow Auto Scaling?

No. Auto scaling is not supported in this release.

Is API Runtime Call Traffic Sent from the Gateway to Management Service?

No. If the identity management (IDM) system lacks the library necessary for local gateway authentication, the gateway calls the IDM system to authenticate users.

What Tools are Available to Assist with the Design and Creation of REST, SOAP, and Other APIs?

To simplify API development, Oracle API Platform Cloud Service - Classic integrates with Oracle Apiary. You can design APIs in either API Blueprint or Swagger 2.0. Interactive documentation is auto generated and the Oracle API Platform Cloud Service - Classic user interface includes a console for accessing the documentation and making API calls. Oracle Apiary also instantiates a mock service which can be used to interact with the examples provided in the API specification file.

How is Documentation Created and Reviewed in the API User Portal?

If the API is configured to use the documentation provided through Oracle Apiary, the Oracle Apiary documentation viewer is available inside the Developer Portal. Documentation is automatically uploaded from Oracle Apiary when the page is loaded. The API Manager who configures the Oracle Apiary documentation must have an Oracle Apiary Pro team account. Users wanting to view documentation in the Developer Portal do not require an Oracle Apiary account.

How Do I Import a New Certificate Chain to the Load Balancer?

You can import a new certificate chain using the Oracle Traffic Director console.

SeeSigning in to the Load Balancer Console for Your Instance to access the Oracle Traffic Director console for your instance. See Importing a Certificate in to import the certificate chain.

Before you import a new certificate to Oracle Traffic Director, make sure these guidelines are followed:

- Place the certificates in the file in this order: 1. server certificate, 2. intermediate certificate, and 3. root certificate.
- Include all certificates in the chain.
- Be sure to select **Certificate Chain** from the **Certificate Type** field on the Import Certificate dialog.

How Do I Configure Keystores on a Gateway Node?

See Configure Keystores in WebLogic Server Administration Console Online Help for information about configuring keystores on a gateway node.

How Do I Obtain a CA-Signed Certificate for the Management Server OTD?

You can obtain a CA-signed certificate in Oracle Traffic Director. See Obtaining a CA-Signed Certificate in the *Oracle Traffic Director Administrator's Guide* for more information.

What Are the Prerequisites for Installing a Gateway Node in Production Mode?

If set the `gatewayExecutionMode` property is set to Production mode, ensure that the OTD public certificate is CA signed. See Obtaining a CA-Signed Certificate and Installing a Certificate in *Oracle Traffic Director Administrator's Guide* to import the

certificate chain. In addition, ensure that the intermediate and root certificate of the CA-signed certificate installed on OTD is trusted by the trust store configured on the gateway. It is also recommended that the gateway should be configured with custom identity and custom trust or custom identity and Java standard trust. See Configure Keystores for WebLogic Server.