webMethods API Portal Administrator's Guide

Version 10.5

October 2019

WEBMETHODS

This document applies to webMethods API Portal 10.5 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: YAP-AG-105-20220406

Table of Contents

About this Guide

This guide describes how you can use webMethods API Portal and other webMethods components to effectively manage APIs for services that you want to expose to consumers, within your organization or outside to partners and third parties. In addition to describing the API management components and workflow, the guide explains configuring API Portal for use with CentraSite, webMethods Mediator, and API Gateway and how to manage API Portal and its users, and how to manage APIs published to API Portal.

To use this guide effectively, you should have an understanding of the APIs that you want to expose to the developer community and the access privileges you want to impose on those APIs.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires credentials for Software AG's Product Support

site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Overview

Why do Organizations Expose APIs?

Organizations often lack the resources to support mobile Bring Your Own Device (BYOD), supply chain, or eCommerce initiatives. By opening a set of APIs to external developers, organizations can reduce costs, expand the reach of their products or services, and create new channels of revenue in the following ways:

- Mobile application developers can create mashups and apps that satisfy a particular user niche and are optimized for specific mobile device types and platforms.
- Enterprise application developers can leverage APIs to simplify integration with suppliers and B2B partners.
- The involvement of external developers fosters innovation and collaboration throughout the development community. In return, the resulting developed applications offer the organization additional potential revenue as those applications reach new markets or customers in new ways.

Why do APIs Need to be Managed?

The APIs that an organization exposes contain core assets the organization would want to protect. As with the services they support, these APIs have a life cycle, need to be managed and governed, and require mediation and security at run time.

From an API provider's perspective, an API management tool is needed that enables the provider to do the following:

- Maintain an inventory of APIs and their associated resources.
- Publish, secure, and retire APIs according to defined service level agreements.
- Onboard API developers and give those developers the ability to publish APIs on behalf of the organization.
- Onboard API consumers who use the published APIs in their own applications.
- Provide tiered access to APIs, for example according to authorization level.
- Track key performance indicators (KPIs) to help monitor and interpret API use.

From an API consumer’s perspective, an API management tool should provide the ability to:

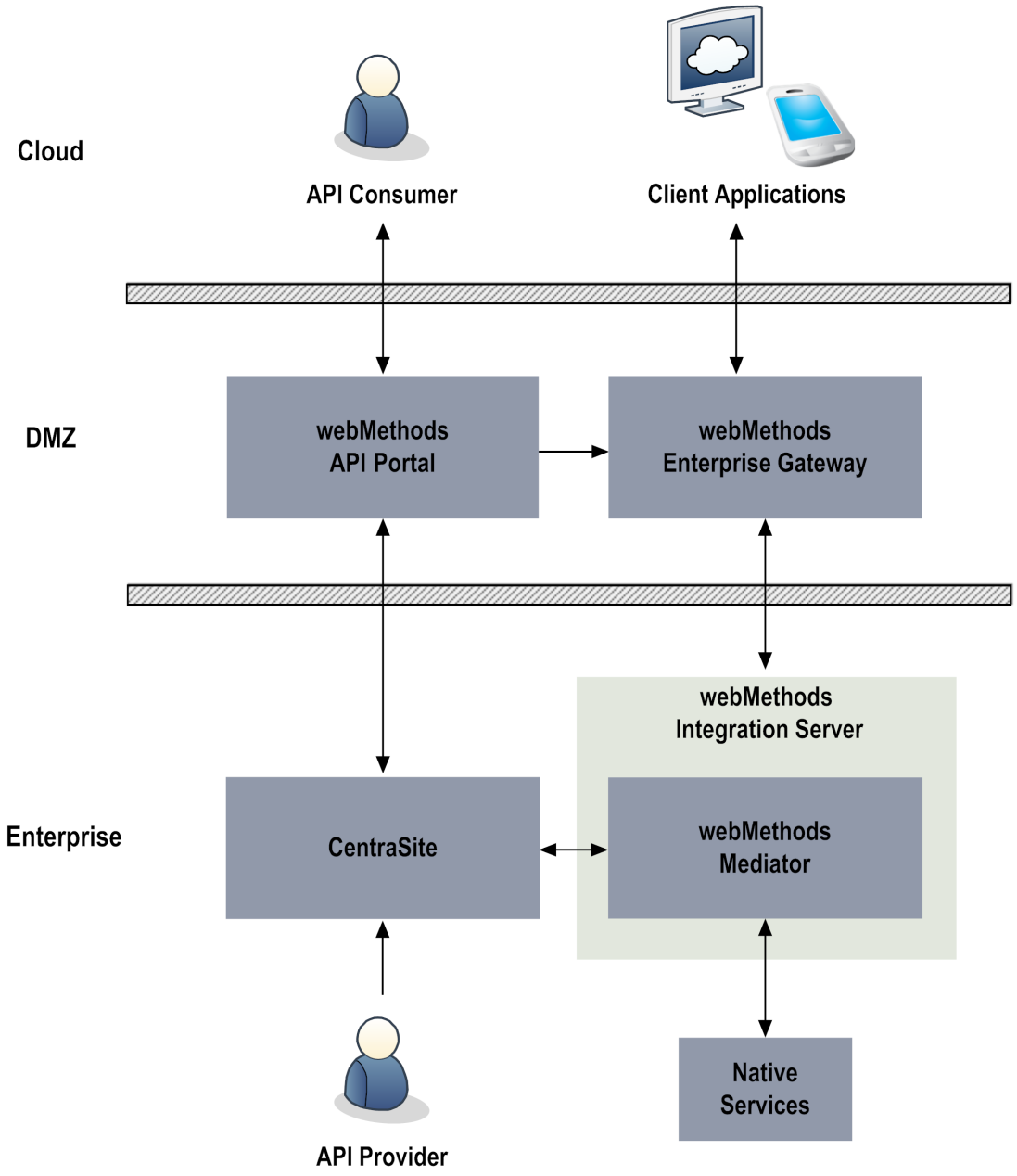
- Browse a catalog of APIs and obtain details and code samples for a specific API.
- Sign up and request and manage access tokens to download an API and its associated resources and documentation.
- Test the functionality of an API.
- Collaborate with other API consumers by way of forums or integration with social media.

What is webMethods API Portal?

webMethods API Portal is a web-based, self-service portal that enables an organization to securely expose APIs to external developers, partners, and other consumers for use in building their own applications on their desired platforms. API Portal provides the following features:

- **Branding and customization.** API Portal administrators can customize their portal's logo, colors, and fonts to match their organization's corporate identity. Administrators can further customize their portal by modifying pages, incorporating widgets, and changing the appearance and organization of APIs in the gallery for easier discovery. For example, APIs in a large catalog can be grouped by business domain, free versus paid, or public versus B2B partner. APIs can also be flagged based on maturity level (for example, beta versus production or release).
- **Support for SOAP and REST APIs.** API Portal supports traditional SOAP-based APIs as well as REST-based APIs. This support enables organizations to leverage their current investments in SOAP-based APIs while they adopt REST for new APIs.
- **Quick, secure providing of access tokens.** Approval workflows simplify the provisioning of API keys and OAuth2 credentials. These workflows enable the API provider to individually approve access token requests that developers submit from API Portal.
- **Easy discovery and testing of APIs.** Full text search capabilities help developers quickly find APIs of interest. API descriptions and additional documentation, usage examples, and information about policies enforced at the API level provide more details to help developers decide whether to adopt a particular API. From there, developers can use the provided code samples and expected error and return codes to try out APIs they are interested in, directly from within API Portal, to see first-hand how the API works.
- **Quick, secure onboarding of new users.** Easy to configure approval workflows in API Portal graphical user interface to define how the user onboarding should take place, with or without confirmations.
- **Community support.** API Portal provides a collaborative community environment where API consumers can rate APIs and contribute to open discussions with other developers.
- **Built-in usage analytics.** API Portal provides dashboarding capabilities. API Portal Administrator, API Providers, and API Consumers can access the dashboard that are visible to them based on their roles to view KPIs based on API Portal page views and API views by users, lifecycle and access token events for an API, monitor the subscriptions per package, and access token requests per API, track total number of logins, active sessions, number of consumers, the success and failure of logins, user registrations, and user audit log, study the API's invocations per user and its performance during runtime, study the API invocation trends by response time, success and failure rates, and track the total API requests over a period of time, requests over time per API, and API request log. This information helps you understand how the APIs are being used, which in turn can help identify ways to improve users' portal web experience and increase API adoption.

The following diagram illustrates a typical scenario of products that make up the webMethods API management product suite.



In this scenario, webMethods API management suite products include the following:

- **webMethods API Portal**. In API Portal, API consumers browse the catalog of APIs that a provider has published. When the consumer finds an API of interest, the consumer can sign up and request an access token to download the API for further investigation and testing.

  API providers who have an API administrator role in API Portal can also view dashboards containing details about API run-time usage.

  Provided with each API Portal installation is a sample portal called SAGTours. The SAGTours sample provides an end-to-end scenario using CentraSite, webMethods Mediator, and API

Portal to demonstrate how the fictitious company SAGTours has customized the content as well as the look and feel of an out-of-the-box API Portal.

- **CentraSite**. CentraSite provides a registry and repository for APIs and offers complete design-time governance of those APIs. API providers add APIs to CentraSite by defining the APIs and their associated resources as objects. When APIs are ready to be made available to consumers, API providers publish the APIs from CentraSite to API Portal.

  CentraSite administrators do the following API management tasks:

  - Register instances of API Portal.
  - Manage API provider and API consumer user accounts.
  - Manage the API catalog.
  - Deploy virtualized APIs to webMethods Mediator.
  - Configure policies to be enforced at run time.
  - Manage API and OAuth2 keys and API access tokens.

  When API providers add API services to CentraSite as assets, the providers can attach supporting documents to the API assets. Examples of such documents include input files containing WSDL or schema definitions, programming guides, sample code, legal notices and terms of use, and associated contracts and plans. When the APIs are published from CentraSite to API Portal, these supporting documents are published to API Portal as well.

- **webMethods Mediator**. Mediator provides complete run-time governance of APIs published to API Portal. Mediator acts as an intermediary between service consumers and service providers. Mediator also enforces access token and operational policies such as security policies for run-time requests between consumers and native services. Using Mediator, API providers can do the following:

  - Enforce security, traffic management, monitoring, and SLA management policies.
  - Transform requests and responses into expected formats as necessary.
  - Perform routing and load balancing of requests.
  - Collect run-time metrics on API consumption and policy evaluation.

- **webMethods Integration Server**. Integration Server hosts Mediator and initiates connections to webMethods Enterprise Gateway. Integration Server also orchestrates the services and provides the connection to back-end systems.

- **webMethods Enterprise Gateway**. Enterprise Gateway protects the APIs on Mediator and other webMethods products installed behind the firewall from malicious attacks initiated by external client applications. Administrators can secure traffic between API consumer requests and the execution of services on Mediator by doing the following:

  - Filter requests coming from particular IP addresses and blacklist specified IP addresses.
  - Detect and filter requests coming from particular mobile devices.

- Avoid additional inbound firewall holes through the use of reverse invoke.

- **webMethods API Gateway**. API Gateway enables an organization to securely expose APIs to external developers, partners, and other consumers for use in building their own applications on their desired platforms. It provides a dedicated, web-based user interface to perform all the administration and API related tasks from the API creation, policy definition and activation, creation of applications, and API consumption. API Gateway gives you rich dashboard capabilities for API Analytics. APIs created in API Gateway can also be published to API Portal for external facing developers' consumption. API Gateway supports REST-based APIs, SOAP-based APIs, and WebSocket APIs, provides protection from malicious attacks, provides a complete run-time governance of APIs, and information about gateway-specific events and API-specific events.

2 Configuring API Portal

Security Considerations

Use the following information to ensure your API Portal installation is protected.

Securing Client Requests

webMethods API Portal supports both HTTP and HTTPS, allowing it to listen on an HTTP port for non-secure client requests and an HTTPS port for secure requests.

Unlike HTTP, HTTPS provides for secure data transmission. HTTPS does this through encryption and certificates. Without HTTPS, unauthorized users might be able to capture or modify data, use IP spoofing to attack servers, access unauthorized services, or capture passwords.

By default, the API Portal load balancer component is set to allow both unencrypted HTTP and encrypted HTTPS/SSL access. Software AG recommends using HTTPS to ensure a secure connection, and disabling the HTTP port.

For instructions on how to disable the HTTP port, see "Disabling a Port" on page 23.

Preventing Use of the HTTP OPTIONS Method

The OPTIONS request method, while part of the HTTP standard, has the potential for allowing incoming requests to obtain information about API Portal server capabilities or to get information about resources, even though the request does not specify a resource action or retrieve a resource.

By default, the API Portal load balancer component is set to allow HTTP OPTIONS method requests. Software AG recommends deactivating the OPTIONS method in the load balancer, preventing it from responding to the requests.

To deactivate the OPTIONS method

1. Stop the loadbalancer component from the API Portal Cloud Controller (ACC).

2. In a text editor, open the `httpd-custom.conf` and the `http-custom-ssl.conf` files from the following directory: *`Software AG_directory`*`\API_Portal\server\bin\work\work_loadbalancer_s\httpd\conf\extra`.

3. Add the following lines to the files:

```
<Location "/">
  <Limit OPTIONS>
    Deny from all
  </Limit>
</Location>
```

4. Start the loadbalancer component from the ACC.

Implementing Secure Password Policies

If the API Portal default password policy does not comply with your security requirements, you can change the password policy settings.

To change the password policies

1. Start a web browser and go to the API Portal User Management Console (UMC) application at: http://*host*:*port*/umc
2. Click **Configuration**.
3. Select **Password policy > General** in the left navigation pane.

   The current password policy settings are shown.
4. Click the edit icon to modify the properties and modify the properties as needed.

   To see a description of each property, hover your cursor over the property name.

Sending Email Notifications

API Portal can send email messages to notify administrators and users about important events and to convey status information.

API Portal can send user management related email messages to notify users about:

- Status of access token requests, renewals, and expiration
- Critical events
- User registration status, including approval workflow notifications

API Portal can also reply to user requests for forgotten passwords.

To enable API Portal to send email notifications, you have to register your SMTP server and set the sender's email address.

Configuring the SMTP Mail Server Connection for API Portal using the User Management Component

You can customize your system configuration to meet your requirements at runtime without having to restart the system. You carry out this part of the configuration in the User Management component. You must have the technical configuration function privilege.

To register the SMTP mail server and set the sender's email address using the user management component

1. Log on to the **User Management Component**.

   http://*host*:*port*/umc

2. Click **Configuration**.

3. Select **SMTP > Connection** in the left navigation pane.

4. Type the SMTP mail server address, including the domain or type the Host name.

   For example: `API_Portal@MyCompany.com`

5. Type the port number of SMTP server. For example, 25.

6. Define the sender's email address.

   You configured the SMTP Mail Server Connection using the User Management Component.

Configuring the SMTP Mail Server Connection for API Portal using ACC

To enable API Portal to send email notifications, you need to register your SMTP server and set the sender's email address.

To register the SMTP mail server and set the sender's email address using ACC

1. Set the SMTP mail server.

   a. Start the ACC.

   b. At the command line, run the `register` external service, where *`smtp_server`* is the SMTP mail server address, including the domain:

   ```
   register external service smtp host=smtp_server port=25
   ```

   c. Verify the setting by entering the following command, and examining the output to see the email server is listed:

   ```
   list external services
   ```

Configuring Ports

API Portal listens for requests on ports that you specify. Each port is associated with a specific type of protocol: HTTP, HTTPS, or email.

By default, the API Portal load balancer component is set to allow both unencrypted HTTP and encrypted HTTPS/SSL access. API Portal has the following pre-configured ports:

Port Type	Default Port Number	Description
HTTP	18101	Unsecured/unencrypted port
HTTPS	18102	Secure/encrypted port
Email	25	SMTP port

API Portal accepts port connection requests as soon as it receives them. If you want to temporarily prevent API Portal from accepting requests on one of its ports, you can disable that port. This action blocks incoming requests from reaching the API Portal server. When a port is disabled, clients receive an error message when they issue requests to it. Later, you can enable the port. If you stop and restart API Portal, the port remains disabled until you enable it. Disabling a port is a convenient way to eliminate developer access to an API Portal once it goes into production.

Disabling a Port

Disabling a port allows you to stop the port from accepting connections or dispatching more requests. For example, you may need to temporarily disable ports for testing, or disable the HTTP port and use only the HTTPS port for secure client requests. To disable a port, use the ACC `reconfigure` command and set the port to 0.

To disable a port

1. Start the ACC.
2. Stop the load balancer component.
3. At the ACC command prompt type:

```
reconfigure loadbalancer_s  +HTTPD.port=0
```

4. Start the load balancer component.
5. To verify that you have deactivated the port, try logging in. The welcome screen is not visible, if you have deactivated the port.

Enabling a Port

When you are ready to have API Portal begin accepting requests on a port you previously disabled, you must enable it. To enable a port, use the ACC `reconfigure` command and set the port number.

To enable a port

1. Start the ACC.

2. Stop the load balancer component.

3. At the ACC command prompt type:

```
reconfigure loadbalancer_s  +HTTPD.port=port_number
```

4. Start the load balancer component.

Testing for HTTPS Requests

To test whether your server is listening to HTTPS requests on the port you specified, bring up your browser and type `https://localhost:`*port* . If the port is working properly, you will see the logon screen for API Portal. If API Portal does not display, check to see if a service running on the machine is listening to the same port.

Considerations for Machines with Multiple Network Interfaces

By default, the load balancer is set at installation time with the server host name and port details. These details are persisted in all load balancer configurations. If you are configuring multiple machines (for example, cloud instances) and have cloned one to many, you need to reconfigure the load balancer for each machine. This ensures each machine has its own identity, and prevents problems at startup.

Reconfiguring the Load Balancer when you configure multiple API Portal machines

If you are configuring multiple API Portal machines, you need to check the load balancer components on each one and ensure that the `HTTPD.servername` parameter is set to the correct IP address of server.

To reconfigure the load balancer component

1. Start the ACC.

2. At the command line, type `show instance loadbalancer_s` to see the current configuration settings of the load balancer component. For example:

```
ACC+ localhost>show instance loadbalancer_s
ID: loadbalancer_s state:STARTED type:com.aris.runnables.httpd.httpd-run-prod-98.0.0

classifier=runnable type=zip
Configuration parameters:
        HTTPD.LimitRequestFieldSize=32768
        HTTPD.access.root=granted
        HTTPD.keepalive=on
        HTTPD.modjk.max_packet_size=32768
        HTTPD.modjk.stickySessions.abs=true
        HTTPD.modjk.stickySessions.ads=true
```

```
HTTPD.modjk.stickySessions.ecp=true
HTTPD.modjk.stickySessions.processboard=true
HTTPD.modjk.stickySessions.umc=true
HTTPD.port=18101
HTTPD.servername=192.0.2.11
HTTPD.ssl.port=18102
appcontext.abs=abs
appcontext.cop=/
appcontext.ecp=collaboration
plugin.ping.search.for.processes=false
zookeeper.application.instance.host= portal01.my.org
zookeeper.connect.string=localhost:18043
```

3. Examine the HTTPD.servername parameter. If it needs to point to another server IP, you can change it by using the `reconfigure` command. To do so:

   a. Stop the load balancer component:

      ```
      stop loadbalancer_s
      ```

   b. Change the value of the HTTPD.servername parameter and specify the new IP address:

      ```
      reconfigure loadbalancer_s +HTTPD.servername=new_IP
      ```

   c. *Optional.* If you want to change the HTTPS port, you have to reconfigure both load balancer instances with the following command:

      ```
      reconfigure loadbalancer_m HTTPD.redirect.http.port =":<port#>"
      HTTPD.redirect.https.port =":<new port #>" HTTPD.servername=<server name>
      ```

   d. Start the load balancer component:

      ```
      start loadbalancer_s
      ```

Reconfiguring the Loadbalancer in case of a DMZ or Reverse Proxy Setup

For on-premise deployments, the recommendation is to place API Portal behind a reverse proxy in the DMZ. The reverse proxy is required for tunneling of requests for the API Portal. The reverse proxy can either be an arbitrary reverse proxy like an Apache with mod_proxy or mod_rewrite modules, a third party security appliance, or a wM Enterprise Gateway (for the pure API traffic).

The API Portal load balancer by default redirects any clients to its fully-qualified host name. For example, if the server's host name is **myportal.mycompany.com**, and a user who is in the domain **mycompany.com** can access the server with its non-qualified name **myportal**, then the user (or rich client and so on being used) is redirected to **myportal.mycompany.com**.

When API Portal is placed behind a reverse proxy in the DMZ the redirect may not work properly with certain network configurations where there are internal and external IPs and no hostname. This can be solved by reconfiguring the Loadbalancer settings.

You have to reconfigure the following parameter settings:

- `HTTPD.servername` parameter to manually tell the load balancer the proper external hostname of the reverse proxy (or IP address) to be used to redirect.
- `HTTPD.RewriteEngine` parameter to disable the automatic redirection.

The loadbalancer registers itself as a service in zookeeper. This registration includes registration of a port, a hostname, and a scheme. This information can then be used by other API Portal applications to generate URLs that are passed to users (for example, in email notifications). In the reverse proxy case, the hostname used for registration should be the external (reverse proxy) address. This has to be configured by reconfiguring the `zookeeper.application.instance.host` parameter.

Reconfiguring the Loadbalancer Settings

1. Start the ACC.
2. At the command line, type `show instance loadbalancer_s` to see the current configuration settings of the load balancer component.

   ```
   ACC+ localhost>show instance loadbalancer_s
   ```

3. Stop the load balancer component.

   ```
   stop loadbalancer_s
   ```

4. Reconfigure the following loadbalancer settings for redirect as required:
   - `HTTPD.servername` parameter to redirect to the proxy server IP. You can change it by using the reconfigure command as follows:

     ```
     reconfigure loadbalancer_s +HTTPD.servername=<reverse_proxy_Address>
     ```

   - Disable the automatic redirection using the following command:

     ```
     reconfigure loadbalancer_s +HTTPD.RewriteEngine=off
     ```

5. Change the value of the zookeeper.application.instance.host parameter and specify the proxy server IP address:

   ```
   reconfigure loadbalancer_s zookeeper.application.instance.host
   ="<reverseproxy_address>"
   ```

6. Start the load balancer component:

   ```
   start loadbalancer_s
   ```

Adding a SSL Certificate to Load Balancer

Pre-requisites:

- Make sure the SSL certificate is available.
- By default, the load balancer is set to allow both unencrypted HTTP (port 18101) and encrypted HTTPS/SSL (port 18102). For SSL, the certificate must fit to the load balancer's host name.
- Ensure the SSL certificate is signed by a Certificate Authority (CA). You can also obtain SSL certificate from an official CA.

To add a SSL certificate to load balancer:

1. Bundle the `extension.key` file used to encrypt an information sent back to the client and the `extension.crt` file into a zip folder.
2. Copy the zip folder to a local directory accessible by the ACC console.
3. Start the ACC console.
4. Stop the load balancer runnables.

   ```
   stop loadbalancer_m
   ```

5. Enhance the loadbalancer_<s, m, or l> with the sslCertificate local file from the location `c:\\temp\\lbcert.zip`

   ```
   enhance loadbalancer_m with sslCertificate local file "c:\\temp\\lbcert.zip".
   ```

   Alternatively, use forward slashes. For example: `c:/temp/lbcert.zip`.

6. Start the load balancer runnables.

   ```
   start loadbalancer_m
   ```

Updating the SSL Certificate

You can update the SSL certificate used by API Portal load balancer through the ACC console.

To update the SSL certificate

1. Start the ACC.
2. Enhance the loadbalancer_s with the sslCertificate local file from the location `C:/WildcardSSLCerti.zip`

   ```
   enhance loadbalancer_s with sslCertificate local file <certf_dir>
   ```

3. Stop the load balancer component:

   ```
   stop loadbalancer_s
   ```

4. Start the load balancer component:

```
start loadbalancer_s
```

API Portal License File

API Portal cannot be viewed or used without a license. If the license was not provided during installation, you must import it before you use API Portal.

Importing the API Portal License File using the ACC

To import the API Portal license file from the ACC

1. Start API Portal if it is not already running.
2. Start the API Portal Cloud Controller (ACC).
3. Ensure all runnables are started. To do so, issue the `list` command.
4. Type the following command:

```
invoke enhancement_importLicense on apiportalbundle_s tenant.name=<tenant>
enhancement.path=" <licensepath> "
```

5. Type `stopall` to stop API Portal.
6. Type `startall` to restart API Portal.

For instructions on how to perform the steps above, see:

- "Starting API Portal (Windows)" on page 88
- "Starting API Portal (Linux/UNIX)" on page 88
- "Stopping API Portal (Windows)" on page 89
- "Stopping API Portal (Linux/UNIX)" on page 90

Importing the API Portal License File using the UMC

To import the API Portal license file from the UMC

1. Start API Portal if it is not already running.
2. Start a web browser and go to the API Portal UMC application at: `http://<host>:<port>/umc`

3. Log in as system user (default password `manager`).
4. In the UMC, click **Licenses** and then import the license file.
5. Open ACC Console and type `stopall` to stop API Portal.
6. Type `startall` to restart API Portal.

For instructions on how to start and stop API Portal, see:

- "Starting API Portal (Windows)" on page 88
- "Starting API Portal (Linux/UNIX)" on page 88
- "Stopping API Portal (Windows)" on page 89
- "Stopping API Portal (Linux/UNIX)" on page 90

Configuring User Registration

When API Portal visitors decide to use the APIs there, they need to have full access as validated users to log in to the portal. To log in to the API Portal, users can either:

- Register with API Portal by providing an email address and password. Upon approval, API Portal creates an account for the user in the UMC, and the user can log in with the email address and password provided at registration. Users can manage their own account details and change the password from the Profiles link in API Portal.
- Use the social account credentials, for example, Google or Facebook. Upon approval, API Portal notifies the user. At the first login, API Portal creates an account for the user in the UMC. Users who log on to API Portal using social accounts can cannot modify anything except their language preference from the **Profiles**page.

Depending on your security requirements, you can choose the level of approval needed when registration requests and social logins arrive:

- **Approval required.** This option is used to define a multi-level approval process to onboard a user. API consumers who would register to the API Portal must be approved by the multiple level of approvers specified in this section. So, you can define more than one level of approval and provide the names of users or user groups that must approve the new user registration at each level.

  Note:
  Only the users with API Administrator or API Provider privileges can approve user registration. The API Consumers cannot be a part of the user groups that approve user registrations.

  When you provide a user group as approver at a level, you can also specify whether a member from the specified user group can onboard the user or all members of the user group must approve the new user. Once the registration is approved by the first level, it is forwarded to

the next level, and so on till it is approved by all levels before being onboarded. The user registration request and the approval at each level is notified to the approvers over email.

Note:
Approval workflows in API Portal are separate from the approval workflows that are used with run-time policies in CentraSite.

- **Email confirmation (default).** Email confirmation provides a simple way to register new users. Upon receiving a user registration request, API Portal sends an email to the requester at the email address provided at registration. The requester clicks the link in the email to activate the user account and the user credentials.
- **Automatic registration.** If it's not essential to review each user registration request, you can use automatic registration, where API Portal automatically processes all user registration requests or social log in requests upon receipt. With automatic registration, API Portal creates the user account and notifies the requester by email that their account is activated and ready to use. The requester needs to log in to the portal using the email address or social account credentials that were provided at registration.

By default, API Portal stores all user registration approvals and email notifications. Depending on the volume of user registration activity for your portal, you may want to periodically purge the approval and email notification entries. For more information, see "Purging Logs" on page 219.

Configuring Approval Workflow for User Registration

Before you configure the registration process, you need to:

- **Customize email notification templates.** Use the default text provided or customize the text as needed. You can use pre-defined tokens as placeholders for specific information. For more information, see "Customizing Email Templates" on page 37.
- **Configure the SMTP mail server.** If you have not already done so, configure the SMTP server to enable API Portal to send email notifications. For instructions, see "Configuring the SMTP Mail Server Connection for API Portal using ACC" on page 22
- **Configure social accounts.** If you want to allow requesters to use a social account to access the portal, you need to configure that access. For instructions, see "User Registration in API Portal with Social Login" on page 38.

When a requester clicks **Register** on the API Portal landing page or clicks **Log in** and types the social login credentials, a registration request appears in the approver's Pending Approvals page. If a requester submits another request while the first request is still pending, API Portal automatically rejects the second request. By default, users with the API Administrator role can approve or reject user registration requests. To include more users to the approval process, add the users to the pre-defined approver group, **API User Registration Approvers**. Or, create a new group and add the required approvers to the group and add the new group to the approval workflow.

To use approval workflows you need to:

1. Assign users as approvers by adding them to the approver group. Determine who in your organization reviews user registration requests, and approve or reject them. In the UMC, add users, who are to review and approve or reject registration requests, to the approver group.

2. Define the workflow approval process. In API Portal, specify that you want to require approval for all registration requests, and select at which points in the approval process. You can specify a user or a group of users to review the registration approval requests.

   API Portal automatically approves registration in the following situations:

   - The **API User Registration Approvers** group is not configured. For example, if the **API User Registration Approvers** group has been renamed or deleted.
   - There are no users in the **API User Registration Approvers** group.

Assigning Users to the Approver Group

API Portal provides a pre-defined approver group, **API User Registration Approvers**. To specify which users receive user registration requests to approve or reject, add the users in the approver group.

Important:
Do not change the name of the API user registration approver group. The name must be **API User Registration Approvers**.

To add or remove users as approvers

1. Start a web browser and go to the API Portal UMC application at: `http://host:port/umc`

2. Click **User management**, and then click **User groups**.

3. Click the **API User Registration Approvers** user group name, and then click **Associated users**.

4. Click **Edit assignment**. Add or remove users as approvers as follows:

   - **To add users:** Select the users you want to add from the **Available items** box, and then click **Add**. The selected users are transferred to the **Assigned items** box, and can now receive and approve user registration requests.
   - **To remove users:** Select the users you want to remove from the **Assigned items** box, and then click **Remove**. The selected users are moved to the **Available items** box, and can no longer receive registration requests.
   - **To add or remove all users at once:** Click **Add all** or **Remove all**.

5. Click **Save**.

Configuring the Multi-Level Approval Workflow

You can configure more than one user or user groups that must approve user registration. You can also specify whether an email notification must be sent to the requests and approver at each level of the registration approval process, the subject, and the content in the email.

To configure the approval workflow

1. In API Portal, click **Administration > User registration**.
2. Click **Approval required**.
3. Select the communities to which the newly onboarded users must be assigned to in the **Enter default communities for newly onboarded users** field.

   By default, the users are assigned to the Public Community.

   The newly onboarded users are a part of the selected communities.
4. Provide one of the following:
   - User name of the approver for who must approve the user registration.
   - Name of the user group who must approve the user registration. Select one of the following options:
     - **Anyone** to complete the registration, if any one member of the group approves the registration.
     - **Everyone** if the registration must be approved by all members of the selected user group.

   If you are providing a user group that is not predefined in API Portal, ensure that the user group is assigned with the webMethods API Portal Viewer privilege. For information on assigning privileges for a user group, see "Assigning Privileges to User Groups" on page 108.
5. Click  to add another level of approver.
6. Perform Steps 4 and 5 till you add the required levels of approvers.
7. Select the notifications that you want to send for each workflow step that you want to use.

   For each step, the **Subject** and **Content** fields contain the content that is used for all notifications sent from API Portal. Use the default content or change the content, as required. For more information, see "Email Notifications Templates and Tokens" on page 37.
8. Click **Apply**.

User Registration Process

You can define a user registration process in one of the three ways:

- user registration through email confirmation
- user registration through an approval by an administrator
- automatic user registration

Email confirmation required

Select this option for user registration through an email confirmation and select the communities to which the newly onboarded users must be assigned to in the Enter default communities for newly onboarded users field. By default, the users will be assigned to the Public Community.

Field	Description
Subject	Is used for providing the subject of the confirmation email.
Contents	Is used for providing the text of the confirmation email.
Apply	Is used for activating the settings.

Approval required

Select this option for user registration through multi-level approval and select the communities to which the newly onboarded users must be assigned to in the Enter default communities for newly onboarded users field. By default, the users will be assigned to the Public Community. For steps to configure the multi-level approval process, see "Configuring the Multi-Level Approval Workflow" on page 32.

Field	Description
Send notification to requester that the request has been received	Activates or Deactivates depending on whether a pending notification is sent to the requester.
Subject	Is used for providing the subject of the pending notification, that is sent to the requester.
Contents	Is used for providing the text of the pending notification, that is sent to the requester.
Send notification to approver(s) that a registration request has been received	Activates or Deactivates depending on whether a pending notification is sent to the approvers.
Subject	Is used for providing the subject of the pending notification that is sent to the approvers.

Field	Description
Contents	Is used for providing the text of the pending notification that is sent to the approvers.
Send notification to requester that the request will be approved	Activates or Deactivates depending on whether an approval completion notification is sent to the requester.
Subject	Is used for providing the subject of the approval completion notification, that is sent to the requester.
Contents	Is used for providing the text of the approval completion notification, that is sent to the requester.
Send notification to requester that the request was rejected	Activates or Deactivates depending on whether a rejection notification is sent to the requester.
Subject	Is used for providing the subject of the rejection notification that is sent to the requester.
Contents	Is used for providing the text of the rejection notification that is sent to the requester.
Send notification to approvers about result of registration request	Activates or Deactivates depending on whether a notification about the registration result is sent to the approvers.
Subject	Is used for providing the subject of the registration result notification that is sent to the approver.
Contents	Is used for providing the text of the registration result notification that is sent to the approver.
Apply	Is used for activating the settings.

Automatic Registration

Select this option for automatic user registration and select the communities to which the newly onboarded users must be assigned to in the Enter default communities for newly onboarded users field. By default, the users will be assigned to the Public Community.

Field	Description
Subject	Is used for providing the subject of the confirmation email.
Contents	Is used for providing the text of the confirmation email.
Apply	Is used for activating the settings.

Pending Approvals

Users who are assigned the API Administrator role and users included in the approver group use the Pending Approvals page in API Portal to view and approve or reject registration requests.

The various actions you can perform in this page are listed in the following table.

Actions	Description
Approve	Approves the onboarding of the selected users.
Reject	Rejects the onboarding of the selected users.
User ID	Displays the user ID of the user who sent a user registration request to API Portal.
E-mail address	Displays the email address of the user who sent a user registration request to API Portal.
Reason for request	Displays the reason the user, who sent a user registration request to API Portal, wrote in the request email.
Origin	Displays the origin of the user who sent a user registration request to API Portal.
Organization	Displays the organization of the user sent a user registration request to API Portal.

Displaying Pending Approvals for User Registration Requests

To either approve or reject the user registration requests you have to navigate to the pending approvals page to see a list of pending approvals.

Prerequisite

You should be a member of the approval user group.

To display pending approvals for user registration requests

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Pending approvals**.

All your pending approvals for the user registration are displayed.

Approving a User Registration Request

As a member of the approver group you can approve a user registration request pending for registration.

Prerequisite

You should be a member of the approver user group.

To approve a user registration request

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Pending approvals**.
3. Select the users you want to approve.
4. Click **Approve**.

You approved users for registration.

If the email notification for approvers is activated, you will receive an email whenever a user requests a registration. Click the link within the e-mail, log on to API Portal and approve the user.

Rejecting a User Registration Request

You should be a member of the **API User Registration Approvers** user group.

To reject a user registration request

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Pending approvals**.
3. Select the users you want to reject.
4. Click **Reject**.

You rejected users from registering.

View and Manage Users

After a user is approved, a user account is created for the user in the UMC. User account details are available from the Profiles link after logging in to API Portal.

- If the user registered with an email address and created a password, API Portal creates the user account when the registration request is approved.
- If the user chooses to log in using a social account (Facebook, Google), API Portal creates the user account after the user logs on for the first time as a registered user. Users must use their social account to view their profile.

Note:

Social account passwords cannot be changed through the UMC or API Portal UI.

From the UMC, an administrator can view and change the user details, such as the email address and phone number, groups the user is assigned to, and privileges assigned to the user. You can also delete a user and its associated account.

Email Notifications Templates and Tokens

API Portal provides default email templates that you can customize as needed. The templates are used in the Define user registration page in API Portal. The templates also use tokens that you can use as placeholder information.

Customizing Email Templates

For each type of registration process on the Define registration process page, there are email templates provided that API Portal uses to send email notifications to requesters about the status of their requests. You can change the default subject and content for any of the email templates.

To change the content of email templates

1. In API Portal, click **Administration > User registration**.
2. Click to select the type of registration process.
3. Edit the **Subject** and **Content** fields as required.
4. Click **Apply**.

Email Tokens

You can use any of the following tokens within your email templates. The tokens are valid in both the email **Subject** and **Content** fields. Before sending the email notification, API Portal replaces the token with the corresponding value.

Token	Is replaced with...
`@requestor.name`	Name of the requestor.
`@loadbalancer.url`	URL of API Portal.
`@recipient.name`	Name of the recipient.
`@tenant.id`	Name of the tenant.
`@recipient.email`	Email id of the recipient.
`@activationlink.ttl`	Time To Live (in minutes) for the email activation link.

Token	Is replaced with...
`@recipient.username`	Username of the recipient.
`@approver.name`	Username of the approver.

User Registration in API Portal with Social Login

By default, API Portal asks new users to register by providing a valid email address and a password. Upon approval, the user logs in to the portal using the email address and password. But you can also enable users to access the portal through a social login. Giving users access with their existing Google or Facebook account means they do not have to register or remember another set of credentials—they simply log in to the portal using their social account. API Portal authenticates a user by accessing their social account.

Social login is a form of single sign-on using existing login information from a social network to sign in to a third-party application. Before an application can access private data of a social media user, it must obtain an access token that grants access to the OAuth provider API.

Social login works with all API Portal registration approval processes. After being approved or clicking on an email confirmation link, users can access the portal. Users who are rejected or who do not have a valid email confirmation are denied access.

When you allow social login to API Portal:

- At the user's first login, API Portal stores the user's social login information, if authorized by the user.
- Users who access the portal with their social account can not change their user profile information or password from the API Portal Profiles link. All user profile fields, with the exception of the **Language** field, are read only, and there is no password change link. Instead users need to go to their social account and make changes there.
- Users can delete their API Portal account from the API Portal Profiles link.
- Dashboards in API Portal can capture and track which social app users access the portal with.

After access with a social account is configured, valid users see a login dialog where they can sign in to API Portal with their social account credentials if they are not already logged in to their social account.

What is OAuth?

OAuth is a standard for authorization that enables client applications to securely access resources on behalf of a resource owner. OAuth specifies processes that allow resource owners to authorize third-parties to access their resources without having to share credentials. OAuth allows an authorization server to issue access tokens to third party clients with the approval of a resource owner or end user. The client can then use the access token to access protected resources offered by the server. OAuth is most commonly used to allow users to log in to a web site using their

Google, Facebook, Twitter, or any other social media account, without worrying about their credentials being compromised.

There are several ways to request an access token from the provider. The process used by API Portal is described below.

1. The user clicks the **Sign in with *social_network*** link on the API Portal login screen.
2. The application creates an authorization URL for the requested provider and redirects the user to that URL.
3. If the user is already logged in to the social network, he is redirected back to the API Portal landing page where he is already logged in based on the approval process defined.
4. If the user is not already logged in, he is offered the possibility to log in at the OAuth provider. After logging in, the user is prompted to grant the permissions requested by API Portal. This process is called *user consent*. If the user gives consent, the OAuth provider redirects the user back to API Portal including a temporary code. If the user does not give consent, the OAuth provider returns an error.
5. After API Portal obtains an access token, it uses the permitted API to determine the identity of the user, and creates a user account in the UMC, and finally logs in the user.

Configuring Google Login

To enable users to log in to the portal through Google, you must first create a Google app and then configure API Portal to access Google account information to authenticate users.

Note:
If you do not already have a Google account, you have to create one. For complete information about Google sign-in, see the documentation available on developers.google.com.

To use Google for user registration and log in

1. Log in to developers.google.com and create an app for API Portal registration.
   a. Go to the Google Developer Documentation.
   b. Log in using your Google account credentials.
   c. Click **Create Project**. Provide the project name and project id, and then click **Create**.
   d. Click on the link for your project. In the left side navigation, go to **APIs & auth > Credentials**, and then click **Create new Client ID**.
   e. Select the application type **Web application**. The warning says that a product name needs to be set. Click **Configure consent screen** and set the product name.

f. At the Create Client ID screen, the application requires that the domain be public. Provide the URL and the relevant port of your website in the **Authorized Javascript Origins** field, and provide the redirect URL and the relevant port in the **Authorized Redirect URLs** field. The URLs must be HTTPS. Click **Create Client ID**.

g. In the left side navigation, go to **APIs & auth > Credentials**. You see the client id and client secret on the right.

h. Note the client key and the client secret from the Google app, because you require it in the next step.

i. In the left side navigation, go to **APIs & auth > APIs**. Search for Google+ API and enable it. It is available in the **Enabled APIs** section.

2. Log in to UMC as Administrator, and configure the OAuth settings for the Google app as described in "OAuth Properties for Social Login" on page 41.

3. If you have not already done so, choose the level of approval needed, as described in "Configuring User Registration" on page 29.

Configuring Facebook Login

To enable users to log in to the portal through Facebook, you must first create a Facebook app and then configure API Portal to access Facebook account information to authenticate users.

Note:
If you do not already have a Facebook ID, you have to create one. For complete information about Facebook login, see the documentation available on developers.facebook.com.

To use Facebook for user registration and log in

1. Log in to developers.facebook.com and create an app for API Portal registration.

   a. Go to http://developers.facebook.com

   b. Log in using your Facebook account credentials.

   c. Click **My Apps** and select **Add a New App**.

   d. In the Create a New App ID dialog box, provide the **Display Name**, **Contact Email**, and select a category **Apps for Pages**.

   e. Click **Create App ID**.

   f. Click **Settings**.

g. Provide **App Domains**.

h. Click **+ Add Platform**.

i. Select **Website** and provide the **Site URL**.

j. Click **Save Changes**.

k. Click **App Review**.

l. For **Your app is in development and unavailable to the public**, select Yes.

m. Click **Confirm** in the confirmation dialog box.

Once the configuration is complete, you can get the **App ID** and **App Secret** in the Settings section.

2. Log in to UMC as Administrator, and configure the OAuth settings for the Facebook app, as described in "OAuth Properties for Social Login" on page 41.

3. If you have not already done so, choose the level of approval needed, as described in "Configuring User Registration" on page 29.

OAuth Properties for Social Login

If you are using social logins to provide access to your API Portal, you need to configure OAuth settings in UMC to authorize the portal to work with the social apps. Log on to UMC as Administrator, and then click the Configuration tab. Click **OAuth** in the left navigation pane. This displays the **General** and **Advanced settings** of OAuth properties that can be set. Click ✎ to modify the required properties. Alternatively, you can also edit these properties by clicking **All** in the left navigation pane and selecting ✎ for the required property.

com.aris.umc.oauth

com.aris.umc.oauth.active
Indicates whether OAuth is used for authenticating portal access from social logins. Set the value to `true` to enable users to log in with their social account. Set the value to `false` to disable social login, and restrict access to valid users with an account in the UMC. The default is `false`.

com.aris.umc.oauth.api.keys
A comma-separated list of API keys obtained from each social app provider used for login. The order of the values specified in this property should match the order of the values specified in the com.aris.umc.oauth.providers property.

com.aris.umc.oauth.api.secrets

A comma-separated list of API secrets obtained from each social app provider used for login. The order of the values specified in this property should match the order of the values specified in the com.aris.umc.oauth.providers property.

com.aris.umc.oauth.debug
Specifies whether debug level output is provided for OAuth operations. Set the value to `true` to enable detailed debug output. Set the value to `false` to disable debug output, and not provide detailed information. The default is `false`.

com.aris.umc.oauth.providers
A comma-separated list of OAuth providers for each social app used for login. Both values are optional. The list of providers specified here defines how many login possibilities are displayed. If, for example, only Google is configured, then the login page displays only **Login with Google**.

com.aris.umc.oauth.tenant
Specifies the default tenant used for social authentication. This value is read-only.

Sample Configuration for Google

In this example, we want to see full debug information for Google logins on the default tenant:

```
com.aris.umc.oauth.active=true
com.aris.umc.oauth.debug=true
com.aris.umc.oauth.providers=facebook,google
com.aris.umc.oauth.api.keys=facebook_key,google_key
com.aris.umc.oauth.api.secrets=facebook_secret_ID,google_secret_ID
com.aris.umc.oauth.tenant= default
```

Removing Social Login

If you no longer want to allow social login from any social account, you need to disable that access in the UMC. Doing so means that users can only register for an account with a valid email address and password.

To remove social registration and login access

1. Log in to UMC as Administrator, and disable OAuth as follows:

   a. Go to the API Portal UMC application at: http://*host*:*port*/umc, and log in with your administrator credentials.

   b. Click the **Configuration** tab.

   c. Select **OAuth > General** in the left navigation pane.

   d. Clear the **Use OAuth** check box.

2. Log out from the UMC.

Configuration Settings

You can configure the configuration settings tenant-wide from this page. The various configuration parameters are described in the following table.

Parameter	Description	Example for values
Automatic Registration for LDAP users	Defines the LDAP user registration settings.	selected (default)
Delete user registration requests once the approval process is complete	Defines whether a user registration request is deleted, when the approval process is completed. If this setting is not selected the user registration request is not deleted after it is approved.	selected (default)
Enable third party approval	Enables external third-party approval for the users onboarded in API Portal. If this setting is selected, then an event is generated in the third-party application when an user registration is approved from API Portal. Administrators can use the generated event to approve the user registration.	cleared (default)
Time To Live (in minutes) for the activation link in minutes	Defines the time interval in minutes for which the email activation link remains active.	30 (default)
Subject of the e-mail sent to the administrator to inform about a failed user request	Subject of the email that is sent to the administrator in case of a failed user request.	API Portal failure :[@request]
Message of the e-mail sent to the administrator to inform	Text that is sent to the administrator in case of a failed user request.	Hello @recipient.name, A request from user (@requestor.name) could not be processed.

Parameter	Description	Example for values
about a failed user request		The details of the request are as follows - API Portal: @loadbalancer.url/#@tenant/home - Tenant: @tenant - Requesting user: @user - Request: @details - Time stamp: @timeStamp - Further details: @reason Best Regards, API Portal Team*** This notification was sent automatically. Do not reply to this email.***
Subject of the email sent to consumers to inform about a failed user request	Subject of the email that is sent to the consumer in case of a failed user request.	Request could not be processed :[@request]
Message of the e-mail sent to consumers to inform about a failed user request	Text that is sent to the consumer in case of a failed user request.	Hello @recipient.name, Your request could not be processed. The details of the request are as follows - API Portal: @loadbalancer.url/#@tenant/home - Request: @request - Time stamp: @timeStamp Try again after some time.Best Regards, API Portal Team*** This notification was sent automatically. Do not reply to this email.***
Enable analytics for global, user, and API audit	Defines, whether collection of analytics should be enabled for global, user and API audit data.	Selected (default)
Enable 'Audit' function for requesting access tokens	Defines, whether auditing should be enabled for an access token request. A log entry is generated when this is selected.	Selected (default)
Enable 'Audit' function for renewing access tokens	Defines, whether auditing should be enabled for an access token renewal. A log entry is generated when this is selected.	Selected (default)
Enable 'Audit' function for revoking access tokens	Defines, whether auditing should be enabled for an access token revoke. A log entry is generated when this is selected.	Selected (default)
Enable 'Audit' function for publishing/creating APIs	Defines, whether auditing should be enabled for APIs being published. A log	Selected (default)

Parameter	Description	Example for values
	entry is generated when this is selected.	
Enable 'Audit' function for republishing/updating APIs	Defines, whether auditing should be enabled for APIs being republished. A log entry is generated when this is selected.	Selected (default)
Enable 'Audit' function for unpublishing/deleting APIs	Defines, whether auditing should be enabled for APIs being deleted. A log entry is generated when this is selected.	Selected (default)
Enable 'Audit' function for purging logs	Defines, whether auditing should be enabled while purging the logs. A log entry is generated when this is selected.	Selected (default)
Subject of the e-mail generated when events occur in API Portal	Subject of the e-mail that is sent for events generated inside portal.	New ${event.type}!!
Message of the e-mail generated when events occur in API Portal	Text of the email that is sent for events generated inside portal.	Hello, There is a new ${event.type} event received from webMethods API Portal (${event.portalURL}). Details of the caller as follows: firstname: ${event.executor.firstname} lastname: ${event.executor.lastname} id: ${event.executor.id} email: ${event.executor.email} Details of the event context as follows: contextual object ref: ${event.source.id} contextual object external ref: ${event.source.externalref} Details of the other context info as follows: <#list event.contextdata?keys as prop> ${prop} = ${event.contextdata[prop]} </#list> Best Regards,

Parameter	Description	Example for values
		API Portal Team *** This notification was sent automatically. Do not reply to this email.***
Notify consumers of access token generation	Specifies whether the consumer should be notified when an access token is generated.	Not selected (default)
Subject of the e-mail sent to consumers to inform about access token generation	Subject of the email that is sent to the consumer when access token is generated.	Access token '@applicationName' generated
Message of the e-mail sent to consumers to inform about access token generation	Text of the email that is sent to the consumers when access token is generated.	Hello @recipient.name, Your request for an access token has been processed. The details of the token is as follows Application name: @applicationName Application description: @applicationDesc APIs: @apis API Key API access key: @apiKeyString Expiry date: @APIKey.expiryDate OAuth2 Credentials Client ID: @clientId Client secret: @clientSecret Best Regards, API Portal Team *** This notification was sent automatically. Do not reply to this email.***
Subject of the e-mail sent to consumers to inform about shared application	Subject of the email that is sent to the consumers when an application is shared with them.	@app.name is shared

Parameter	Description	Example for values
Template of the e-mail sent to consumers to inform about shared application	Text of the email that is sent to the consumers when an application is shared with them.	Hello @recipient.name, @requestor.name has shared the @app.name application with you! Please log on to API Portal to view its details by clicking: @loadbalancer.url/#@tenant.id/applications. Best regards, API Portal Team *** This notification was sent automatically. Do not reply to this email.***
Subject of the e-mail sent to consumers to inform about unshared application	Subject of the email that is sent to the consumers when the access to a shared application is revoked.	Access to @app.name is revoked
Template of the e-mail sent to consumers to inform about unshared application	Text of the email that is sent to the consumers when the access to a shared application is revoked.	Hello @recipient.name, @requestor.name has revoked your access to the "@app.name" application. Best Regards, API Portal Team *** This notification was sent automatically. Do not reply to this email.***
Relative endpoint to be notified when new events occur	Defines the relative endpoints for notifying events	Blank (default) Example: /apimgmt/events
HTTP method to be used to notify events	Defines the HTTP method for notifying events	Blank (default) Example: POST
Payload to be used to notify events	Defines the payload for notifying events	Blank (default) Example schema: `{ "firstname" : "{executor.firstname}", "name" : "{executor.name}", "lastname" : "{executor.lastname}", "email" : "{executor.email}",`

Parameter	Description	Example for values

		<pre>"type" : "{type}"
		}</pre>

Content type to be used to notify events	Defines the content type required for notifying events	Blank (default) Example: application/json
Plug-ins enabled for publication of external events	Defines the plugins enabled for publishing external events	SMTP (default)

My application usage report template	Template of the email that is being used to generate application usage report	Example for values:
		<pre><html>
		<body>
		Hello @recipient.name,
		
		 Your Application usage summary
		for the portal instance
		  @loadbalancer.url/#@tenant.id/
		 Please find the summary of
		your application usage in
		the below table
		
		<#assign table = "background:#FFF;">
		<#assign table_th = "font-size:16px;
		font-weight:400;
		color:#fff;text-align:left;
		padding:20px;
		background-color:#0899CC;">
		<#assign table_tr_td =
		"font-weight:400;
		color:#5f6062;font-size:13px;
		padding:20px 20px 20px 20px;
		border-bottom:1px solid #e0e0e0;">
		<#assign align_center =
		"text-align:center;">
		<table style="${table}">
		<thead style="border-radius(5px);">
		<tr>
		<th style="${table_th}">Name</th>
		<th style="${table_th}">Usage</th>
		</tr>
		</thead>
		<tbody>
		<#list response as app>
		<tr>
		<td style="${table_tr_td}">
		${app.name}</td>
		<td style="${table_tr_td}
		${align_center}">
		${app.usage}</td>
		</tr>
		</#list>
		</tbody>
		</table></pre>

Parameter	Description	Example for values
		   Your Application usage detail <#list response as app> Please find the detailed usage of application : ${app.name} <table> <thead> <tr> <th style="${table_th}">#</th> <th style="${table_th}">Date</th> <th style="${table_th}">API</th> <th style="${table_th}">Resource</th> <th style="${table_th} ${align_center}">Usage</th> </tr> </thead> <tbody> <#list app.details as detail> <tr> <td style="${table_tr_td}">${detail?counter}</td> <td style="${table_tr_td}">${detail.key}</td> <td style="${table_tr_td}">${detail.api}</td> <td style="${table_tr_td}">${detail.resource}</td> <td style="${table_tr_td}">${detail.count}</td> </tr> </#list> </tbody> </table> </#list> Best Regards, API Portal Team *** This notification was sent automatically. Do not reply to this email.*** </body> </html>
API consumption request approver group	Defines which user group is allowed to decide about an API consumption.	API Consumption Approvers

Parameter	Description	Example for values
Ignore API Runtime events if API is not available	Ignores the runtime events for an API if that API is not present.	selected (default)
Ignore API Runtime events if application is not available	Ignores the runtime events for an application if that application is not present.	not selected (default)
Expiration days for events	Defines the time after which the logged events expire.	30d (default)
Expiration days for analytics data	Defines the time after which the analytics data expire.	45d (default)
File formats of supported documents	Defines the supporting documents and file formats that can be attached or uploaded.	.pdf,.doc,.zip,.jpg, .jpeg,.docx,.xls,.xlsx,.png, .ppt,.pptx
Maximum file size (in MB) of supported documents	Specifies the maximum file size for the supporting document or file that can be attached or uploaded.	2 MB (default)
API response attachment types	Specifies the types of API response attachments that are supported by API Portal.	application/octet-stream, application/vnd.mspowerpoint, application/pdf, image/gif, image/png, image/jpeg, image/bmp, image/webp, audio/midi, audio/mpeg, audio/webm, audio/ogg, audio/wav, video/webm, video/ogg
Host names allowed for tryout	Specifies the host name that can be used in Try API section. You can use the host names specified in this field only. If no host name is specified here, you can try an API using any host name.	Blank (default)
Maximum API response payload size for tryout (in KB)	Specifies the maximum size, in KB, of response messages returned by Try API page.	512 KB (default)

Customizing Tenant-wide Configuration

You can customize the configuration tenant-wide from the configurations page.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To customize tenant-wide configuration:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Configuration**.
4. Click **Configuration settings**.
5. Configure the relevant tenant-wide settings.

You customized the tenant-wide configuration settings.

Resetting the Default Configuration Settings Tenant-wide

You can reset the tenant-wide configurations to default values from the **Configuration** page.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To restore the default configuration setting tenant-wide

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Configuration**.
4. Click **Configuration settings**.
5. Click **Reset**. A confirmation dialog is displayed.
6. Click **Reset**.

You tenant-wide configured values are reset to default values.

Advanced Configuration of API Portal

The user registration parameters are configurable from the API Portal user interface.

You can perform advanced configurations from the API Portal user interface from Administration > Configuration > Configuration Settings. You can configure the validity period of e-mail activation link received through e-mail for registration, various e-mail notifications for success and failure for user registrations, and enabling audit log for access token events.

The table lists the advanced configurations in API Portal.

Step	Where to find the procedures
Import the API Portal license file.	" API Portal License File" on page 28
Create a technical user in API Portal and assign the user to the API Provider role. Specify this user when registering an API Portal instance in CentraSite and API Gateway. It is considered a best practice not to tie critical actions (such as publishing data or APIs) to a real user whose credentials can expire. **Note:**Software AG recommends specifying the same user credentials as the technical user created in CentraSite, Mediator and API Gateway.	"Overview of Managing Users" on page 106
Customize the API Portal user registration and the e-mail-templates for user registration.	"Configuring User Registration" on page 29
Define the advanced configurations of API Portal.	"Configuration Settings" on page 43
Customize the user registration with social media for API Portal.	"User Registration in API Portal with Social Login" on page 38
Customize the API Portal branding and set up API Portal templates.	"API Portal Views" on page 228

Accessing Elasticsearch

The Elasticsearch runnable is secured by default. You require a password to access Elasticsearch, if you want access it from anywhere away from API Portal. You can use the auto-generated credentials or provide a user name and password of your preference to access the runnable. You can disable the secured access, if required.

Accessing Elasticsearch Using Auto-generated Access Details

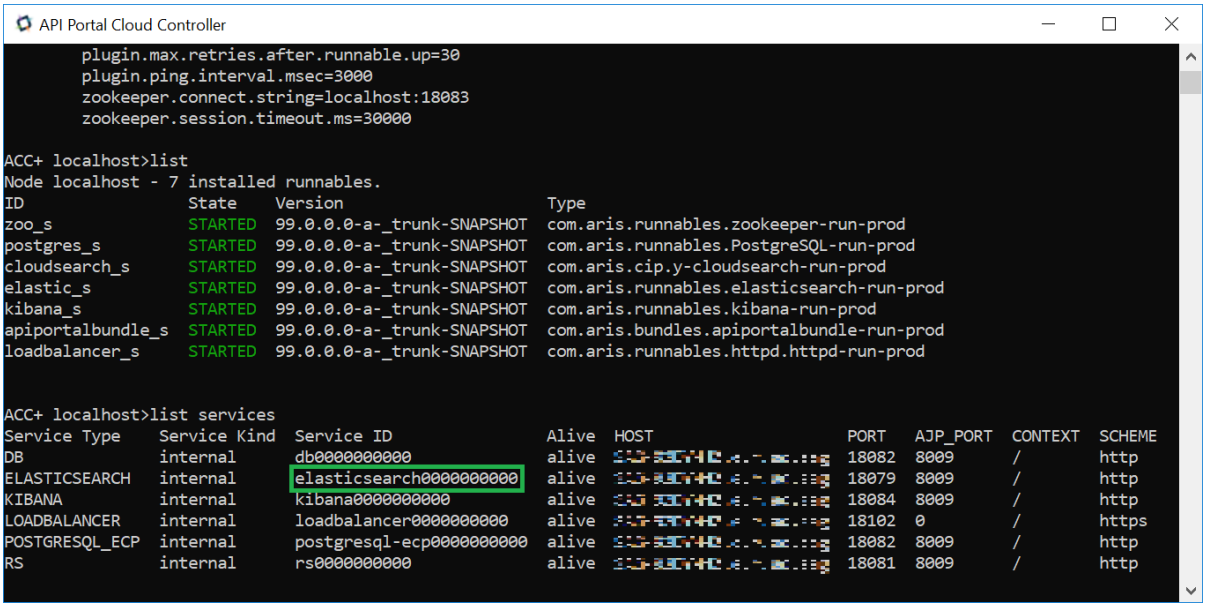
You can access Elasticsearch using an auto-generated username and password.

To access the Elasticsearch runnable

1. Start the ACC.
2. Run the following command:

```
list services
```

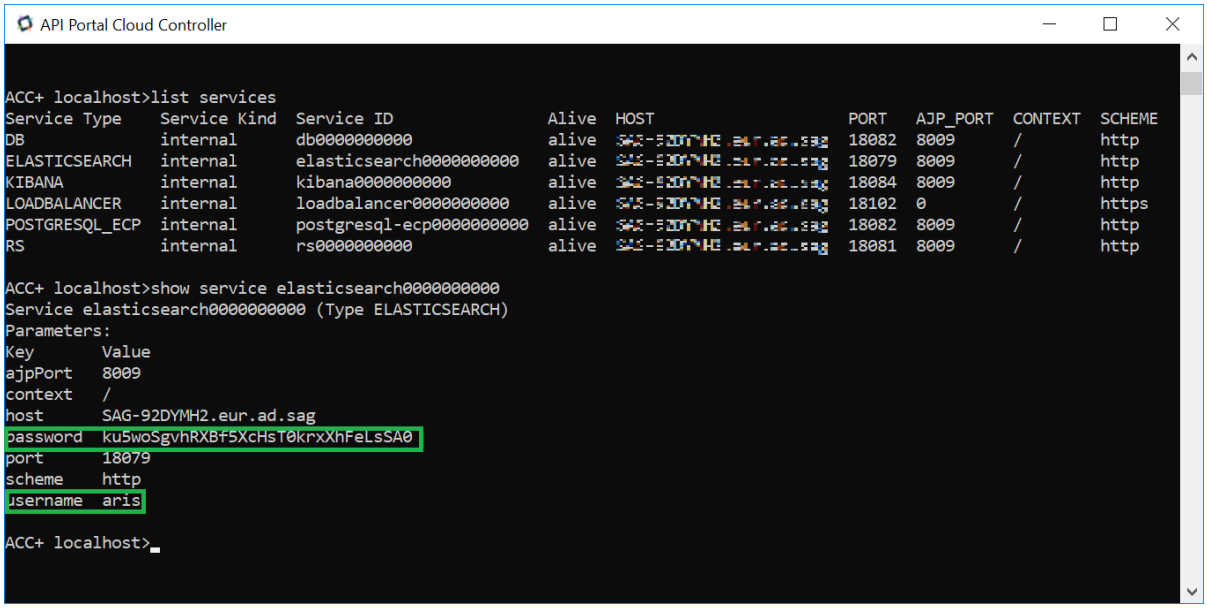
The list of available services is displayed.



3. Copy the Service ID of the Elasticsearch and run the following command using the copied Service ID.

```
show service elasticsearch_serviceid
```

The parameters of the Elasticsearch service are displayed. You can use the username and password to access the secured Elasticsearch instance.



Configuring Elasticsearch Access Details

You can configure the username and password for a secured instance of the Elasticsearch runnable.

To provide username and password

1. Start the ACC.
2. Run the following command with the required username and password in the specified placeholders:

   ```
   reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name=\"username"\
   +ELASTICSEARCH.aris.api.user.password=\"password"\
   ```

   For information on using special characters in ACC commands, see "Usage of Special Characters in ACC Commands" on page 54.

The authentication details that you provided are saved. You can access the Elasticsearch runnable using the username and password obtained from using the show service *elasticsearch_serviceid* command in ACC.

Usage of Special Characters in ACC Commands

If your username or password includes any special character, add a backslash (\) before the character for the ACC to process the special character as a string. That is, if there is an ampersand (&), in your password, add a backslash before the ampersand. For example,

```
reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name=\"admin"\
+ELASTICSEARCH.aris.api.user.password=\"lion\&tiger"\
```

For more information on the usage of ACC commands, run the following commands in ACC:

```
help
```

or to get help on a particular command run the following:

```
help <command>
```

Disabling Secure Access of Elasticsearch

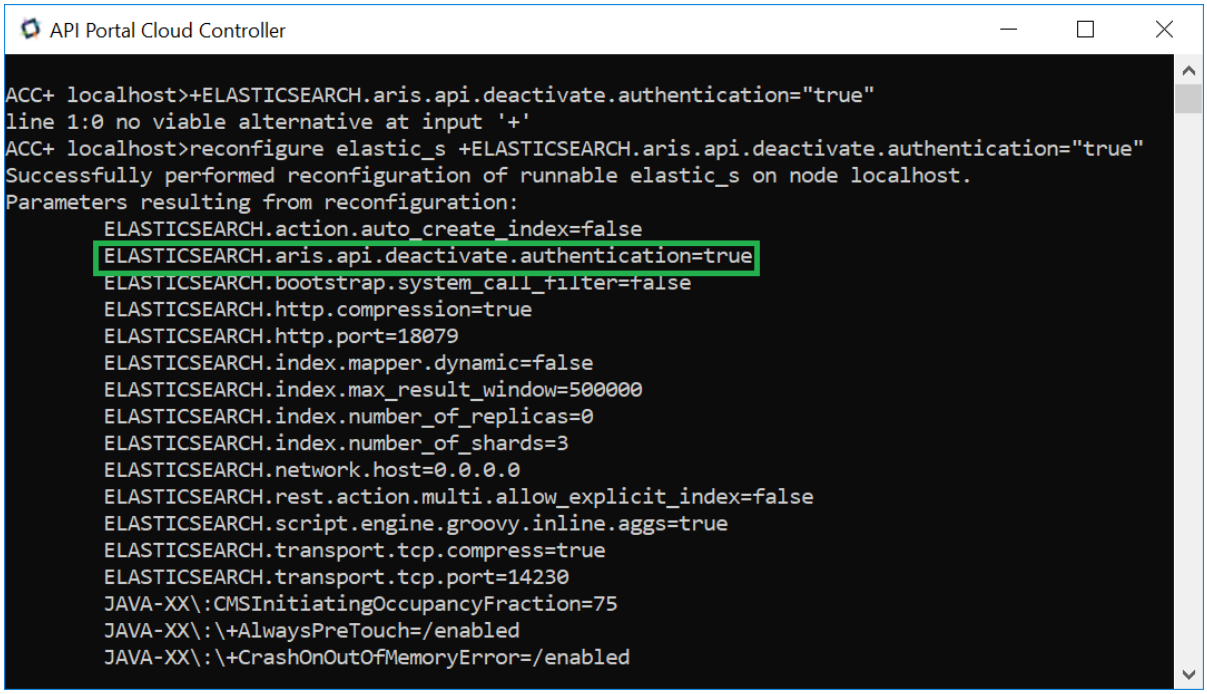
You can disable the secure access of Elasticsearch to allow users to access the runnable without any authentication when it has to be accessed from anywhere away from API Portal.

To disable the secure access of Elasticsearch

1. Start the ACC.
2. Run the following command:

```
reconfigure elastic_s +ELASTICSEARCH.aris.api.deactivate.authentication="true"
```

The Elasticsearch runnable security is disabled.



You can now access the runnable without providing any authentication.

Configuring API Portal with External Databases

API Portal by default uses Postgres database. API Portal can also be configured with the following external databases:

- Microsoft SQL Server 2016

- Microsoft SQL Server 2017
- Oracle Database 11g
- Oracle Database 12c
- Oracle Database 19c

Configure API Portal with Microsoft® SQL Server

To customize the API Portal with Microsoft® SQL Server 2016 and SQL Server 2017, you need the following components:

- An operating Microsoft® SQL Server database.
- The Microsoft® JDBC Driver sqljdbc7.jar. You can download this driver from the Microsoft Web Site to a directory of your choice.
- SQL scripts and all additional files. These scripts can be downloaded from the ARIS Download Center. For API Portal 10.5, you must download the 10SR10 scripts (ARIS10.0.SR10.DatabaseScripts.zip

The SQL scripts creates a database and necessary database objects required by the API Portal components.

Note:
Database scripts should be executed from the machine where sqlcmd client is running.

To configure API Portal with Microsoft ® SQL Server

1. Back up data from API Portal.
   a. Log on to API Portal as an Administrator.
   b. Click  in the right top corner of the API Portal window to display the menu options.
   c. Click **Administration** > **Manage data**.
   d. Select **Backup**.
   e. Select the relevant options and click **Backup**.

      A success message appears when the backup process is completed. The backup file with an extension `.acb` is created and saved in the downloads section. You can move the file and save it in another location of your choice.
2. Open API Portal Cloud Controller and execute the following commands.

```
stopall
deconfigure postgres_s
start zoo_s
```

3. To configure envset.bat, login to the machine where MSSQL server is running and go to the directory where the script files are downloaded from ARIS Download centre. Scripts are present in the folder,
download_root_folder\ARIS.xxx.DatabaseScripts\DatabaseScripts\Design&ConnectServer\mssql

   Note:
   For API Portal 10.5, download `ARIS.10.0.SR10.DatabaseScripts.zip` file.

4. Open the `envset.bat` file, modify the following fields, and save the file:
   - SET MSSQL_SAG_MSSQL_SERVER_NAME=*Server Name (Machine name where MSSQL server is running)*
   - SET MSSQL_SAG_DATABASE_NAME=*Database name (Ex: Portal104DB)*
   - SET MSSQL_SAG_FILEGROUP_FILE_DIR=*MS SQL file path* (for example, C:\msqldata*Database name*
   - SET MSSQL_SAG_APP_USER=*UserName (Application username which will be created. Ex: dbuser)*
   - SET MSSQL_SAG_APP_PWD=*Password (Password of the application user. Ex: dbuser123)*

5. Before running the database scripts ensure that the Microsoft SQL Server client (sqlcmd) is available in the command prompt. Run the `inst.bat` file; this drops the existing schema and creates new schema. Run the following 2 commands, which will create 2 schemas, one for the master tenant and one for the default tenant.
   - `create_schema_for_tenant.bat CIP_MASTER`
   - `create_schema_for_tenant.bat CIP_DEFAULT`

6. Switch to the machine where API Portal is installed. Add the JDBC drivers to API Portal classpath.
   a. Start API Portal Cloud Controller.
   b. Run the following command:

      ```
      enhance apiportalbundle_s with commonsClasspath
      local file "sqljdbc_jar_location"
      ```

      Example:

      ```
      ACC + localhost> enhance apiportalbundle_s with commonsClasspath
      local file "C:/jars/sqlserver/sqljdbc7.jar"
      ```

7. Register the external service database.

   a. In API Portal Cloud Controller, run the following commands:

   ```
   register external service db url="jdbc:sqlserver://
   servername:port;
   DatabaseName=databasename"
   driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
   username="username" password="password"
   maxIdle=15 maxActive=300 maxWait=10000
   removeAbandoned=false removeAbandonedTimeout=600
   logAbandoned=true defaultAutoCommit=false
   rollbackOnReturn=true host=servername jmxEnabled=true
   database.admin.user="database.admin.user"
   database.admin.password="database.admin.password"
   ```

   An external service identifier is returned once the above command is executed, for example, it returns the service id as db0000000000.

   b. Run the following command to assign the service to the default and master tenants:

   ```
   assign tenant default to service db0000000000 com.aris.cip.db.schema=CIP_DEFAULT
   assign tenant master to service db0000000000 com.aris.cip.db.schema=CIP_MASTER
   ```

8. In API Portal Cloud Controller, run the following command to start all the runnables:

   ```
   startall
   ```

9. Restore the backed up data.

   a. Log on to API Portal as an Administrator.

   b. Click in the right top corner of the API Portal window to display the menu options.

   c. Click **Administration** > **Manage data**.

   d. Select **Restore**.

   e. Click **Upload** and select the relevant backup file to be uploaded.

   f. Select the relevant options and click **Restore**.

   A success message appears when the restore process is completed.

Configure API Portal with an Oracle Database

To customize the API Portal with Oracle 11g, Oracle 12c, or Oracle 19c database you need the following components:

- An operating Oracle database.

- ojdbc driver jar, which can be downloaded from Oracle website to a directory of your choice in the machine where API Portal is installed. Example: ojdbc6.jar.

  Note:
  For configuring Oracle 12c and Oracle 19c with API Portal 10.5 or an higher version, download ojdbc8.jar driver.

- SQL scripts and all additional files. These scripts can be downloaded from the ARIS Download Center. For API Portal 10.5, you must download the 10SR10 scripts (ARIS10.0.SR10.DatabaseScripts.zip

The SQL scripts creates a database and necessary database objects required by the API Portal components.

Note:
Database scripts should be executed from the machine where sqlplus client is running.

To configure API Portal with an Oracle Database

1. Take a back up of data from API Portal.
2. Start Zookeeper service.
3. Register external Oracle DBMS.
4. Create database schema for tenants.
5. Enhance the runnables with the JDBC driver for the external database system.
6. Register the external service database.
7. Assign tenant to the database service.
8. Restore the backed up data.

Taking a Back up of API Portal database

1. Log on to API Portal as an Administrator.
2. Click  in the right top corner of the API Portal window to display the menu options.
3. Click **Administration** > **Manage data**.
4. Select **Backup**.
5. Select the relevant options and click **Backup**.

A success message appears when the backup process is completed. The backup file with an extension `.acb` is created and saved in the downloads section. You can move the file and save it in another location of your choice.

Starting Zookeeper Service

Since all runnables use Zookeeper to find services, the information needed to find and use an external DBMS must be added to Zookeeper as well. ACC commands are used for registering and updating external services. With the register external service command you perform the initial registration of the service. With the list external services command you can get an overview of all external services that are currently registered. You can use the following command to review the properties of individual registered services:

```
show external service
```

With the update external service command you can change individual properties of the registered service.

1. Open API Portal Cloud Controller and run the following commands.

```
stopall
deconfigure postgres_s
start zoo_s
```

Registering External Oracle DBMS

You must update the values of variables in the envset.bat files before running the schema creation scripts.

1. Login to the machine where oracle server is running, and go to the directory where the script files are downloaded from ARIS Download centre. Scripts are present in the folder, *download_root_folder*`\ARIS.xxx.DatabaseScripts\DatabaseScripts\Design&ConnectServer\oracle`

2. Open the `envset.bat` file, modify the following fields, and save the file:

   - SET CIP_ORA_BIN_PATH=*Path where sqlplus.exe can be found* (for example C:\app\<username>\product\11.2.0\<dbname>\BIN)
   - SET TARGET_HOST=*DB Server Name (Machine name in which Oracle server is running)*
   - SET TARGET_PORT=*Port (Port in which oracle server is running. Example: 1521)*
   - SET TARGET_SERVICE_NAME= *Service name (Name of the oracle service. Example: XE for oracle 11g)*
   - SET CIP_INSTALL_USER=*User Name (Database administrator username)*
   - SET CIP_INSTALL_PWD=*Password (Database administrator password)*
   - SET CIP_TS_DATA= *Name of the table space in which the application schema data will be stored. By default, the value in this field is DATA. Change the value as USERS.*

- SET CIP_APP_USER=*Username (User that will be used by the application. Example: dbuser)*
- SET CIP_APP_PWD=*Password (Password of the application user. Example: dbuser123)*
- SET CIP_TENANT_SCHEMA_PWD=*Password (Password used for tenant schemas. Example: dbuser123)*

3. Before running the database scripts ensure that the Oracle query tool (sqlplus) is available in the command prompt.

4. Run the envset.bat file.

5. Run cip_create_app_user.bat file. This creates the application user, which was specified in envset.bat file.

   For Oracle 12c and Oracle 19c, the following changes should be made for error free execution of the scripts files,

   a. To avoid `ORA-65096: invalid common user or role name` error during schema creation, open cip_create_empty_tenant_schema.sql and cip_create_app_user.sql files, and add the following after "set verify off", alter session set "_ORACLE_SCRIPT"=true;

   b. If complex password policy is enabled by default in the database and the application user password does not comply to it, an error message displays, while creating the tenant schema. To avoid this, open cip_create_empty_tenant_schema.sql file and add the following after "BEGIN",

```
EXECUTE IMMEDIATE 'ALTER PROFILE default LIMIT
                PASSWORD_VERIFY_FUNCTION null';
```

Creating Database Schema for Tenants

By default, each API Portal installation has at least two tenants namely *default* and *master*. So, you must create schemas for these tenants; and additional schemas for each additional tenant you want to use.

1. To create database schema for the tenants default and master, run the following commands in a command line.

   - `cip_create_schema_for_tenant.bat CIP_MASTER`
   - `cip_create_schema_for_tenant.bat CIP_DEFAULT`

   Note:
   You can use DbVisualizer to ensure that the schemas are created.

Enhancing Runnables with JDBC Driver for External Database

You must add the JDBC drivers to all runnables that access the database by using the commonsClasspath enhancement point. The simplest way to make the JDBC driver available for

the enhancement commands is to copy the file into a subdirectory of the machine that you are using to configure your runnables.

1. From the machine where API Portal is installed, add the JDBC drivers to API Portal classpath.

   a. Start API Portal Cloud Controller.

   b. Run the following command:

   ```
   enhance apiportalbundle_s with commonsClasspath
   local file "location of ojdbc file"
   ```

   Example:

   ```
   ACC+ localhost>enhance apiportalbundle_s with commonsClasspath
   local file "C:/jdbc/jar/ojdbc8.jar"
   ```

Registering the External Service Database

1. In API Portal Cloud Controller, run the following commands:

   ```
   register external service db url="jdbc:oracle:thin:@
   servername:port/servicename"
   driverClassName=oracle.jdbc.OracleDriver jmxEnabled=true maxActive=100
   maxIdle=15 logAbandoned=true rollbackOnReturn=true maxWait=10000
   removeAbandoned=false defaultAutoCommit=false
   username=Application Username password=Application User password
   host=servername
   ```

   An external service identifier is returned once the above command is executed, for example, it returns the service id as db0000000000.

   Note:
   For information on using special characters in ACC commands, see "Usage of Special Characters in ACC Commands" on page 54.

Assigning Tenant to the Database Service

After you register your Oracle database system with API Portal and created the schemas for each tenant, you must specify the following in the API Portal using ACC command explained in this section:

- The database service that should be used for each tenant.
- The name of the schema in the respective database that should be used for each tenant.

1. Run the following command to assign the service to the default and master tenants:

   ```
   assign tenant default to service db0000000000
   com.aris.cip.db.schema=CIP_DEFAULT
   assign tenant master to service db0000000000
   com.aris.cip.db.schema=CIP_MASTER
   ```

2. In API Portal Cloud Controller, run the following command to start all the runnables:

```
startall
```

Restoring the backed up data

Once you have registered the required external database system, and assigned tenants, you can now restore the backed up data.

1. Log on to API Portal as an Administrator.
2. Click in the right top corner of the API Portal window to display the menu options.
3. Click **Administration** > **Manage data**.
4. Select **Restore**.
5. Click **Upload** and select the relevant backup file to be uploaded.
6. Select the relevant options and click **Restore**.

   A success message appears when the restore process is completed.

If you want to create a new tenant you can do the following:

- Run the following commands:
  - Switch to the machine where Oracle server is running. Open sqlplus cmd and execute the following command,

    ```
    cip_create_schema_for_tenant.bat CIP_TENANTNAME
    ```

  - Switch to the machine where API Portal is installed and run the following command in ACC console, assign tenant default to service db0000000000

    ```
    assign tenant TENANTNAME to service db0000000000
    com.aris.cip.db.schema=CIP_TENANTNAME
    ```

  After running the above command, start all runnables.
- Run the following commands in ACC console to create a new tenant:

```
ACC + localhost> create tenant tenant_name
master.tenant.user.name=system master.tenant.user.pwd=manager
ACC + localhost> invoke enhancement_importLicense on apiportalbundle_s
tenant.name=tenant_name local file
enhancement.path="Install_dir/API_Portal/license.xml"
tenant.user.name=system tenant.user.pwd=manager
ACC + localhost> invoke prepareTenant on apiportalbundle_s
tenant.name=tenant_name isDemo=false
```

High Availability setup in API Portal

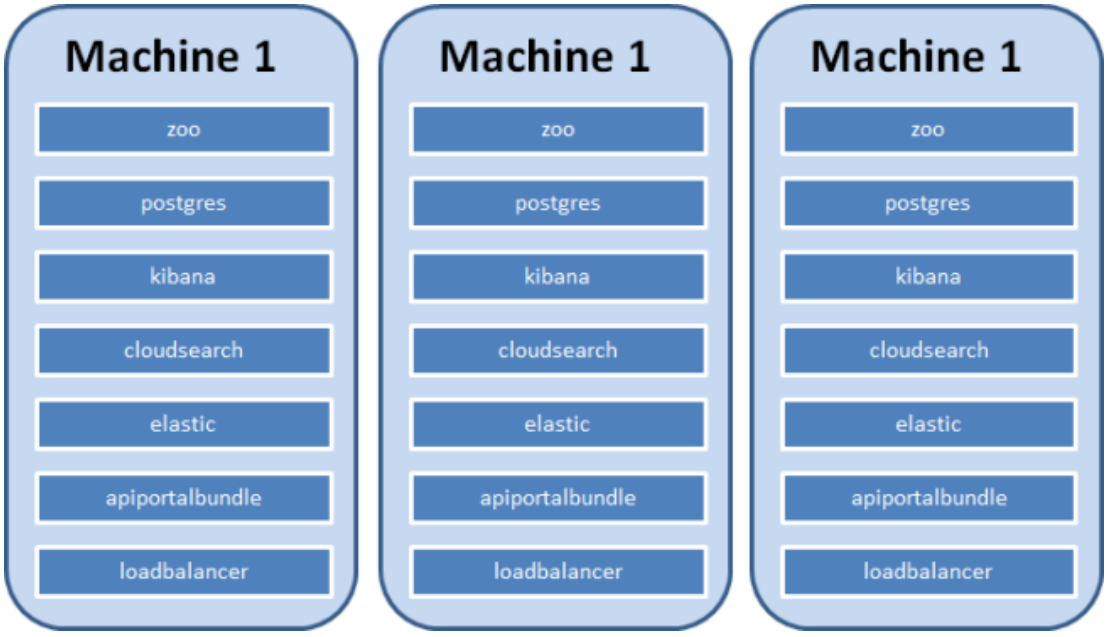
The High Availability(HA) set up in API Portal ensures continuous operation in cases of a fail-over scenario where up to one runnable of each kind or a whole machine becomes unavailable.

Prerequisites:

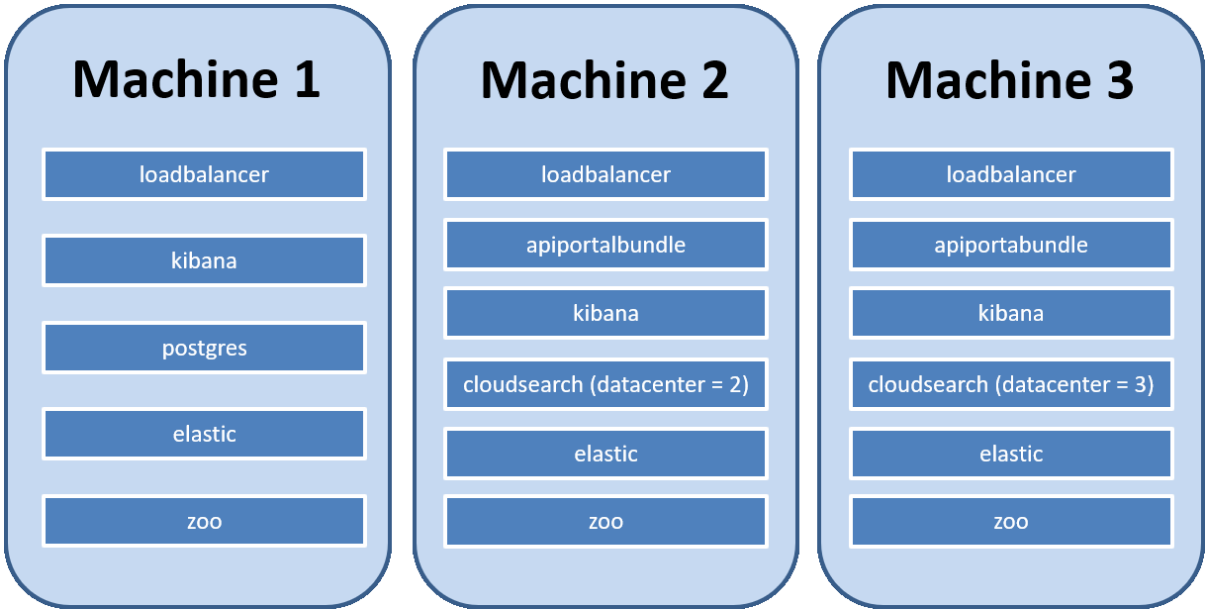
- A minimum of three machines with API Portal set up on each of them. The High Availability setup requires the installation of Zookeeper and Elasticsearch in three nodes. However, the Load balancer, API Portal bundle, and Cloudsearch can be installed in any of two nodes.
- Switch-off firewalls or open appropriate ports on each machine so that they can access all components on the other machines.
- Ensure that the same username and password is provided for all Elasticsearch instances in a cluster.

On installing API Portal, the runnables are installed and are visible in the API Portal Cloud Controller (ACC).

The figure below depicts a general 3-machine API Portal setup with the runnables installed.



The figure below depicts a 3-machine HA set up for API Portal and the distribution of the runnables there after.



Machine 2 and 3 are configured the same way. Either when the full machine goes down or the runnables zoo and elastic stop working, with the runnables on machine 2 and 3 running properly, the system is still in the operation mode.

Setting up API Portal HA setup

This procedure describes in detail the setting up of the HA setup for API Portal.

Prerequisites:

Ensure to follow the below to set up HA and update the fixes.

- Install API Portal but do not start any runnables.

Note:
To install fix on a HA setup:

- Ensure that you install the fixes in all nodes of the cluster using UpdateManager, if you are installing API Portal 10.5 Fix 23 or later.
- Post fix installation, connect to the cluster environment using ACC console and ensure all runnables are in the STARTED state.

To configure API Portal HA set up

- Add Nodes
- Deconfigure Runnables and Reconfigure Zookeeper
- Reconfigure Elasticsearch
- Reconfigure Other Runnables
- Start the High Availability Setup

Note:
The sample commands given in the following steps are wrapped to align with the page margins. Hence, when you copy them it is recommended that you copy the code in a Notepad file, remove the line breaks, and use them in ACC or Command prompt.

Adding of Nodes

To start with the High availability configuration, you must add the nodes in ACC and create the 3-node environment.

1. Add worker nodes to ACC.

   a. Start ACC.

   b. Execute the add node command for each of the worker nodes. The basic syntax of the `add node` command is as follows:

   ```
   add node logicalNodeName ipAddressOrHostname
   @agentPort agentUsername agentPassword
   ```

   Replace *logicalNodeName* with the logical node name you want to assign to that node and with which you will later refer to it in the ACC commands.

   For information on using special characters in ACC commands, see "Usage of Special Characters in ACC Commands" on page 54.

2. Create a 3-node environment.

   a. On machine1, create a nodelist.pt file in the folder `SAGInstallDir\API_Portal\server`, which contains the following lines:

   Tip:
   Replace machinename in the following commands with the respective names or IP addresses of your machines (nodes). For example, *apiportalnode1*.

   add node n1 machinename @18011 Clous g3h31m

   add node n2 machinename @18011 Clous g3h31m

   add node n3 machinename @18011 Clous g3h31m

   set current node n1

   b. Run the following command:

   ```
   SAGInstallDir\API_Portal\server\acc>acc.bat -n
   SAGInstallDir\API_Portal\server\nodelist.pt -c
   SAGInstallDir\API_Portal\server\generated.apptypes.cfg
   ```

   Note:
   In the above command, replace acc.bat with acc.sh if you are using Linux.

This creates an ensemble between the instances in the cluster.

c. To view the 3-node cluster in ACC, run the command:

```
ACC+ n1>list nodes
```

The 3-node cluster has all nodes listed listening on port 18011 using REST services as follows:

n1 : machine1 (18011) OK

n2 : machine2 (18011) OK

n3 : machine3 (18011) OK

Deconfiguring Runnables and Reconfiguring Zookeeper

Once the nodes are setup, you must deconfigure the unnecessary runnables, and start the zookeeper runnable in all three nodes.

1. Cleanup the unnecessary runnables by running the following commands, in ACC, to deconfigure the runnables for all the 3 nodes:

```
ACC+ n1>on n1 deconfigure zoo_s
ACC+ n1>on n1 deconfigure cloudsearch_s
ACC+ n1>on n1 deconfigure apiportalbundle_s
ACC+ n1>on n2 deconfigure zoo_s
ACC+ n1>on n2 deconfigure postgres_s
ACC+ n1>on n3 deconfigure zoo_s
ACC+ n1>on n3 deconfigure postgres_s
```

2. Create a zookeeper cluster in ACC by running the following commands:

```
ACC+ n1>on n1 add zk
ACC+ n1>on n2 add zk
ACC+ n1>on n3 add zk
ACC+ n1>commit zk changes
```

The zoo runnables are started.

3. Run the following command in ACC:

```
list zk instances
```

The following configuration is displayed:

```
3 Zookeeper instances:
Node InstID MyID State     Cl Port Port A Port B Type
n1   zoo0   1    STARTED   14281    14285   14290   Master
n2   zoo0   2    STARTED   14281    14285   14290   Master
n3   zoo0   3    STARTED   14281    14285   14290   Master
```

Reconfiguring Elasticsearch

After starting the Zookeeper runnable, you must reconfigure Elasticsearch to form a Elasticsearch cluser. When reconfiguring, you have to specify same user name and password for the Elasticsearch runnables in all three nodes.

1. Execute the following commands in ACC:

```
ACC+ n1>on n1 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine1
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine2:esTCPport,
machine3:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
ACC+ n1>on n2 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine2
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine1:esTCPport,
machine3:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
ACC+ n1>on n3 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine3
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine1:esTCPport,
machine2:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
```

Note:
In the three commands above, replace machine1, machine2, and machine3 with the names or IP addresses of your machines and esTCPport with Elasticsearch TCP Port. You can retrieve the TCP port of an Elasticsearch runnable by executing the `show runnable elastic_s` command in ACC.

2. To validate the Elasticsearch cluster, execute the below command:

```
validate elasticsearch cluster
```

This displays the following message:

```
Found 3 Elasticsearch instances in one
cluster across all currently registered nodes.
There were no errors.
```

3. To set the same user name and password for the Elasticsearch runnable in all three nodes, execute the below command:

```
on n1 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
on n2 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
on n3 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
```

For example,

```
on n1 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="adminuser"
   +ELASTICSEARCH.aris.api.user.password="adminpassword"
```

Reconfiguring Other Runnables

After the Elasticsearch, you must reconfigure other runnables and specify the order in which they must be started.

1. Reconfigure Kibana runnable on all three nodes as follows:

```
on n1 reconfigure kibana_s -zookeeper.connect.string
on n2 reconfigure kibana_s -zookeeper.connect.string
on n3 reconfigure kibana_s -zookeeper.connect.string
```

2. Reconfigure the PostgreSQL database on n1, so that it knows about all zookeeper cluster members and accepts connections from all locations, by running the following command:

```
on n1 reconfigure postgres_s -zookeeper.connect.string
+postgresql.listen_addresses = "'*'"
```

   The only kind of scaling that is possible with PostgreSQL at the moment is that of scaling across tenants. Currently, the data of a single tenant always resides in a single database instance, so the load created by a single tenant on the database instance needs to be handled by that instance. At the same time, a tenant's database instance is a single point of failure for that tenant; if the tenant's DB goes offline, the tenant becomes unusable until the DB is available again. In particular for production use on mission critical systems, when high availability is of interest, this approach is not an ideal solution. Since API Portal does not support a highly available configuration using our PostgreSQL runnable, you have to use an external DBMS like Oracle or MS SQL, which offer mechanisms for clustering and high availability. For details on configuring external databases, see "Configuring API Portal with External Databases" on page 55.

3. Define two cloudsearch instances, on the nodes n2 and n3, where each one belongs to a different data center:

```
on n2 reconfigure cloudsearch_s -zookeeper.connect.string
+zookeeper.application.instance.datacenter = n2
on n3 reconfigure cloudsearch_s -zookeeper.connect.string
+zookeeper.application.instance.datacenter = n3
```

4. Reconfigure the apiportalbundle runnable on the nodes n2 and n3 as follows:

```
on n2 reconfigure apiportalbundle_s -zookeeper.connect.string
on n3 reconfigure apiportalbundle_s -zookeeper.connect.string
```

5. Reconfigure the loadbalancer on n1 to point to all three zookeeper cluster members as follows:

```
on n1 reconfigure loadbalancer_s -zookeeper.connect.string
on n2 reconfigure loadbalancer_s -zookeeper.connect.string
on n3 reconfigure loadbalancer_s -zookeeper.connect.string
```

API Portal can be accessed through multiple hostnames. If one of the loadbalancer fails, the application can be accessed using the other available loadbalancers. You can ignore this step if external load balancer is not required. If you have an external load balancer Software AG recommends you to place a highly available loadbalancer (HA LB) in front of the loadbalancer runnables. To do this the following has to be added to the loadbalancer runnable configuration:

- `HTTPD.servername` to specify the hostname or IP address of the HA loadbalancer
- `HTTPD.zookeeper.application.instance.port`, the port on which the HA loadbalancer receives the http/https requests
- `zookeeper.application.instance.scheme` which specifies the scheme http/https

For example:

```
on n1 reconfigure loadbalancer_s HTTPD.servername=HTTPD.servername
HTTPD.zookeeper.application.instance.port=PORT
zookeeper.application.instance.scheme=SCHEME
```

6. Change the startup order of the runnables by running the following commands:

```
ACC+ n1>on n1 set runnable.order = "zoo0 < (elastic_s, kibana_s, postgres_s)
< loadbalancer_s"
ACC+ n1>on n2 set runnable.order = "zoo0 < (elastic_s, kibana_s)
< cloudsearch_s < apiportalbundle_s < loadbalancer_s"
ACC+ n1>on n3 set runnable.order = "zoo0 < (elastic_s, kibana_s)
< cloudsearch_s < apiportalbundle_s < loadbalancer_s"
```

Starting the High Availability Setup

1. Create a file `startupScript.bat` under SAGInstallDir\API_Portal\server. Copy the following content into the file.

Note:
If you are using Linux, you must name the file with the following content as `startupScript.sh`.

```
#
# start Zookeeper Ensemble
#
on n1 start zoo0
on n2 start zoo0
on n3 start zoo0
on n1 wait for STARTED of zoo0
on n2 wait for STARTED of zoo0
on n3 wait for STARTED of zoo0
#
# start Elasticsearch Cluster
#
on n1 start elastic_s
on n2 start elastic_s
on n3 start elastic_s
on n1 wait for STARTED of elastic_s
on n2 wait for STARTED of elastic_s
on n3 wait for STARTED of elastic_s
```

```
#
# start Kibana
#
on n1 start kibana_s
on n2 start kibana_s
on n3 start kibana_s
on n1 wait for STARTED of kibana_s
on n2 wait for STARTED of kibana_s
on n3 wait for STARTED of kibana_s
#
# start PostgreSQL Database
#
on n1 start postgres_s
on n1 wait for STARTED of postgres_s
#
# start CloudSearch
#
on n2 start cloudsearch_s
on n3 start cloudsearch_s
on n2 wait for STARTED of cloudsearch_s
on n3 wait for STARTED of cloudsearch_s
#
# start API Portal Bundle
#
on n2 start apiportalbundle_s
on n3 start apiportalbundle_s
on n2 wait for STARTED of apiportalbundle_s
on n3 wait for STARTED of apiportalbundle_s
#
# finally, start loadbalancer
#
on n1 start loadbalancer_s
on n2 start loadbalancer_s
on n3 start loadbalancer_s
on n1 wait for STARTED of loadbalancer_s
on n2 wait for STARTED of loadbalancer_s
on n3 wait for STARTED of loadbalancer_s
```

2. Execute the script by running the following command from the command prompt:

```
SAGInstallDir\API_Portal\server\acc>acc.bat
-n SAGInstallDir\API_Portal\server\nodelist.pt -c
SAGInstallDir\API_Portal\server\generated.apptypes.cfg -f
SAGInstallDir\API_Portal\server\startupScript.bat
```

   In the above command, replace the .bat files to .sh files, if you are using Linux OS. That is, acc.bat and startupScript.bat files must be replaced with acc.sh and startupScript.sh.

3. Ensure the HA setup is successfully running.

   In the default installation, access the User Management Component (UMC) at http://*machine name*/umc, the ARIS Document Storage (ADS) at http://*machine name*/ads, the collaboration component at http://*machine name*/collaboration, and the API Portal at http:/*machine name*.

   The Elasticsearch cluster consist of three nodes. Each index is split in 3 parts, called shards, which are replicated once and distributed over the three available nodes. If one node goes offline, the system can still fill up the complete index making it a fail-over system.

Setting up API Portal HA setup with External Database (Oracle)

This procedure describes in detail the setting up of the HA setup for API Portal with an external Database (Oracle).

Ensure the following before you start configuring the setup:

- Install API Portal but do not start any runnables.
- You have a running Oracle instance.

To setup API Portal HA set up with an external Database

- Add Nodes
- Deconfigure Runnables and Reconfigure Zookeeper
- Reconfigure Elasticsearch
- Reconfigure Other Runnables
- Register External Oracle DBMS
- Create Database Schemas for Tenants
- Enhance Runnables with JDBC Driver for External Database
- Assign Tenant to the Database Service
- Start the High Availability Setup

Note:
The sample commands given in the following steps are wrapped to align with the page margins. Hence, when you copy them it is recommended that you copy the code in a Notepad file, remove the line breaks, and use them in ACC or Command prompt.

Adding of Nodes

To start with the High availability configuration, you must add the nodes in ACC and create the 3-node environment.

1. Add worker nodes to ACC.

   a. Start ACC.

   b. Execute the add node command for each of the worker nodes. The basic syntax of the `add node` command is as follows:

```
add node logicalNodeName ipAddressOrHostname
@agentPort agentUsername agentPassword
```

Replace *logicalNodeName* with the logical node name you want to assign to that node and with which you will later refer to it in the ACC commands.

For information on using special characters in ACC commands, see "Usage of Special Characters in ACC Commands" on page 54.

2. Create a 3-node environment.

   a. On machine1, create a nodelist.pt file in the folder `SAGInstallDir\API_Portal\server`, which contains the following lines:

   Tip:
   Replace machinename in the following commands with the respective names or IP addresses of your machines (nodes). For example, *apiportalnode1*.

   add node n1 machinename @18011 Clous g3h31m

   add node n2 machinename @18011 Clous g3h31m

   add node n3 machinename @18011 Clous g3h31m

   set current node n1

   b. Run the following command:

   ```
   SAGInstallDir\API_Portal\server\acc>acc.bat -n
   SAGInstallDir\API_Portal\server\nodelist.pt -c
   SAGInstallDir\API_Portal\server\generated.apptypes.cfg
   ```

   Note:
   In the above command, replace acc.bat with acc.sh if you are using Linux.

   This creates an ensemble between the instances in the cluster.

   c. To view the 3-node cluster in ACC, run the command:

   ```
   ACC+ n1>list nodes
   ```

   the 3-node cluster has all nodes listed listening on port 18011 using REST services as follows:

   n1 : machine1 (18011) OK

   n2 : machine2 (18011) OK

   n3 : machine3 (18011) OK

Deconfiguring Runnables and Reconfiguring Zookeeper

Once the nodes are setup, you must deconfigure the unnecessary runnables, and start the zookeeper runnable in all three nodes.

1. Cleanup the unnecessary runnables by running the following commands, in ACC, to deconfigure the runnables for all the 3 nodes:

```
ACC+ n1>on n1 deconfigure zoo_s
ACC+ n1>on n1 deconfigure cloudsearch_s
ACC+ n1>on n1 deconfigure apiportalbundle_s
ACC+ n1>on n1 deconfigure postgres_s
ACC+ n1>on n2 deconfigure zoo_s
ACC+ n1>on n2 deconfigure postgres_s
ACC+ n1>on n3 deconfigure zoo_s
ACC+ n1>on n3 deconfigure postgres_s
```

2. Create a zookeeper cluster in ACC by running the following commands:

```
ACC+ n1>on n1 add zk
ACC+ n1>on n2 add zk
ACC+ n1>on n3 add zk
ACC+ n1>commit zk changes
```

   The zoo runnables are started.

3. Run the following command in ACC:

```
list zk instances
```

   The following configuration is displayed:

```
3 Zookeeper instances:
Node InstID MyID State     Cl Port Port A Port B Type
n1   zoo0   1    STARTED   14281    14285   14290   Master
n2   zoo0   2    STARTED   14281    14285   14290   Master
n3   zoo0   3    STARTED   14281    14285   14290   Master
```

Reconfiguring Elasticsearch

After starting the Zookeeper runnable, you must reconfigure Elasticsearch to form a Elasticsearch cluser. When reconfiguring, you have to specify same user name and password for the Elasticsearch runnables in all three nodes.

1. Execute the following commands in ACC:

```
ACC+ n1>on n1 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine1
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine2:esTCPport,
machine3:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
ACC+ n1>on n2 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine2
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine1:esTCPport,
machine3:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
```

```
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
ACC+ n1>on n3 reconfigure elastic_s
+ELASTICSEARCH.node.name = machine3
+ELASTICSEARCH.cluster.name=apiportal
ELASTICSEARCH.discovery.zen.ping.unicast.hosts="machine1:esTCPport,
machine2:esTCPport"
+ELASTICSEARCH.discovery.zen.minimum_master_nodes=2 -zookeeper.connect.string
+ELASTICSEARCH.index.number_of_replicas=1
-ELASTICSEARCH.sonian.elasticsearch.zookeeper.client.host
```

Note:
In the three commands above, replace machine1, machine2, and machine3 with the names or IP addresses of your machines and esTCPport with Elasticsearch TCP Port. You can retrieve the TCP port of an Elasticsearch runnable by executing the `show runnable elastic_s` command in ACC.

2. To validate the Elasticsearch cluster, execute the below command:

```
validate elasticsearch cluster
```

This displays the following message:

```
Found 3 Elasticsearch instances in one
cluster across all currently registered nodes.
There were no errors.
```

3. To set the same user name and password for the Elasticsearch runnable in all three nodes, execute the below command:

```
on n1 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
on n2 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
on n3 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="<username>"
   +ELASTICSEARCH.aris.api.user.password="<password>"
```

For example,

```
on n1 reconfigure elastic_s +ELASTICSEARCH.aris.api.user.name="adminuser"
   +ELASTICSEARCH.aris.api.user.password="adminpassword"
```

Reconfiguring Other Runnables

After the Elasticsearch, you must reconfigure other runnables and specify the order in which they must be started.

1. Reconfigure Kibana runnable on all three nodes as follows:

```
on n1 reconfigure kibana_s -zookeeper.connect.string
on n2 reconfigure kibana_s -zookeeper.connect.string
on n3 reconfigure kibana_s -zookeeper.connect.string
```

2. Define two cloudsearch instances, on the nodes n2 and n3, where each one belongs to a different data center:

```
on n2 reconfigure cloudsearch_s -zookeeper.connect.string
+zookeeper.application.instance.datacenter = n2
on n3 reconfigure cloudsearch_s -zookeeper.connect.string
+zookeeper.application.instance.datacenter = n3
```

3. Reconfigure the apiportalbundle runnable on the nodes n2 and n3 as follows:

```
on n2 reconfigure apiportalbundle_s -zookeeper.connect.string
on n3 reconfigure apiportalbundle_s -zookeeper.connect.string
```

4. Reconfigure the loadbalancer on n1 to point to all three zookeeper cluster members as follows:

```
on n1 reconfigure loadbalancer_s -zookeeper.connect.string
on n2 reconfigure loadbalancer_s -zookeeper.connect.string
on n3 reconfigure loadbalancer_s -zookeeper.connect.string
```

   API Portal can be accessed through multiple hostnames. If one of the loadbalancer fails, the application can be accessed using the other available loadbalancers. You can ignore this step if external load balancer is not required. If you have an external load balancer Software AG recommends you to place a highly available loadbalancer (HA LB) in front of the loadbalancer runnables. To do this the following has to be added to the loadbalancer runnable configuration:

   - `HTTPD.servername` to specify the hostname or IP address of the HA loadbalancer
   - `HTTPD.zookeeper.application.instance.port`, the port on which the HA loadbalancer receives the http/https requests
   - `zookeeper.application.instance.scheme` which specifies the scheme http/https

   For example:

```
on n1 reconfigure loadbalancer_s HTTPD.servername=HTTPD.servername
HTTPD.zookeeper.application.instance.port=PORT
zookeeper.application.instance.scheme=SCHEME
```

5. Change the startup order of the runnables by running the following commands:

```
ACC+ n1>on n1 set runnable.order = "zoo0 < (elastic_s, kibana_s)
< loadbalancer_s"
ACC+ n1>on n2 set runnable.order = "zoo0 < (elastic_s, kibana_s)
< cloudsearch_s < apiportalbundle_s < loadbalancer_s"
ACC+ n1>on n3 set runnable.order = "zoo0 < (elastic_s, kibana_s)
< cloudsearch_s < apiportalbundle_s < loadbalancer_s"
```

Registering External Oracle DBMS

You must update the values of variables in the envset.bat files before running the schema creation scripts.

1. Login to the machine where oracle server is running, and go to the directory where the script files are downloaded from ARIS Download centre. Scripts are present in the folder, *download_root_folder*`\ARIS.xxx.DatabaseScripts\DatabaseScripts\Design&ConnectServer\oracle`

2. Open the `envset.bat` file, modify the following fields, and save the file:

- SET CIP_ORA_BIN_PATH=*Path where sqlplus.exe can be found* (for example C:\app\<username>\product\11.2.0\<dbname>\BIN)
- SET TARGET_HOST=*DB Server Name (Machine name in which Oracle server is running)*
- SET TARGET_PORT=*Port (Port in which oracle server is running. Example: 1521)*
- SET TARGET_SERVICE_NAME= *Service name (Name of the oracle service. Example: XE for oracle 11g)*
- SET CIP_INSTALL_USER=*User Name (Database administrator username)*
- SET CIP_INSTALL_PWD=*Password (Database administrator password)*
- SET CIP_TS_DATA= *Name of the table space in which the application schema data will be stored. By default, the value in this field is DATA. Change the value as USERS.*
- SET CIP_APP_USER=*Username (User that will be used by the application. Example: dbuser)*
- SET CIP_APP_PWD=*Password (Password of the application user. Example: dbuser123)*
- SET CIP_TENANT_SCHEMA_PWD=*Password (Password used for tenant schemas. Example: dbuser123)*

Note:
In the above command, replace the envset.bat to envset.sh, if you are using Linux OS.

3. Before running the database scripts ensure that the Oracle query tool (sqlplus) is available in the command prompt.

4. Run the envset.bat file, if you are using Windows. Else, run the envset.sh, if you are using Linux.

5. Run cip_create_app_user.bat file, if you are using Windows. Else, run the cip_create_app_user.sh file, if you are using Linux. This creates the application user, which was specified in envset.bat or envset.sh file.

   For Oracle 12c and Oracle 19c, the following changes should be made for error free execution of the scripts files,

   a. To avoid `ORA-65096: invalid common user or role name` error during schema creation, open cip_create_empty_tenant_schema.sql and cip_create_app_user.sql files, and add the following after "set verify off", alter session set "_ORACLE_SCRIPT"=true;

   b. If complex password policy is enabled by default in the database and the application user password does not comply to it, an error message displays, while creating the tenant schema. To avoid this, open cip_create_empty_tenant_schema.sql file and add the following after "BEGIN",

```
EXECUTE IMMEDIATE 'ALTER PROFILE default LIMIT
                PASSWORD_VERIFY_FUNCTION null';
```

Creating Database Schema for Tenants

By default, each API Portal installation has at least two tenants namely *default* and *master*. So, you must create schemas for these tenants; and additional schemas for each additional tenant you want to use.

1. To create database schema for the tenants default and master, run the following commands in a command line.

   - `cip_create_schema_for_tenant.bat CIP_MASTER`
   - `cip_create_schema_for_tenant.bat CIP_DEFAULT`

   In the above command, replace the cip_create_schema_for_tenant.bat files to cip_create_schema_for_tenant.sh files, if you are using Linux OS.

   Note:
   You can use DbVisualizer to ensure that the schemas are created.

Enhancing Runnables with JDBC Driver for External Database

You must add the JDBC drivers to all runnables that access the database by using the commonsClasspath enhancement point. The simplest way to make the JDBC driver available for the enhancement commands is to copy the file into a subdirectory of the machine where the apiportalbundle runnable is installed. That is, either node 2 or node 3.

1. Run the following command in the ACC console to add JDBC drivers to API Portal classpath:

   ```
   enhance apiportalbundle_s with commonsClasspath
   local file "location of ojdbc file"
   ```

   Example:

   ```
   ACC + n1>on n2 enhance apiportalbundle_s with commonsClasspath
   local file "DownloadLocation/jar/ojdbc8.jar"
   ```

Registering the External Service Database

1. In API Portal Cloud Controller, run the following commands:

   ```
   register external service db url="jdbc:oracle:thin:@
   servername:port/servicename"
   driverClassName=oracle.jdbc.OracleDriver jmxEnabled=true maxActive=100
   maxIdle=15 logAbandoned=true rollbackOnReturn=true maxWait=10000
   removeAbandoned=false defaultAutoCommit=false
   username=Application Username password=Application User password
   host=servername
   ```

   An external service identifier is returned once the above command is executed, for example, it returns the service id as db0000000000.

Assigning Tenant to the Database Service

After you register your Oracle database system with API Portal and created the schemas for each tenant, you must specify the following in the API Portal using ACC command explained in this section:

- The database service that should be used for each tenant.
- The name of the schema in the respective database that should be used for each tenant.

1. Run the following command to assign the service to the default and master tenants:

```
assign tenant default to service db0000000000
com.aris.cip.db.schema=CIP_DEFAULT
assign tenant master to service db0000000000
com.aris.cip.db.schema=CIP_MASTER
```

Starting the High Availability Setup

1. Create a file `startupScript.bat` under SAGInstallDir\API_Portal\server. Copy the following content into the file.

Note:
If you are using Linux, you must name the file with the following content as `startupScript.sh`.

```
#
# start Zookeeper Ensemble
#
on n1 start zoo0
on n2 start zoo0
on n3 start zoo0
on n1 wait for STARTED of zoo0
on n2 wait for STARTED of zoo0
on n3 wait for STARTED of zoo0
#
# start Elasticsearch Cluster
#
on n1 start elastic_s
on n2 start elastic_s
on n3 start elastic_s
on n1 wait for STARTED of elastic_s
on n2 wait for STARTED of elastic_s
on n3 wait for STARTED of elastic_s
#
# start Kibana
#
on n1 start kibana_s
on n2 start kibana_s
on n3 start kibana_s
on n1 wait for STARTED of kibana_s
on n2 wait for STARTED of kibana_s
on n3 wait for STARTED of kibana_s
#
# start CloudSearch
```

```
#
on n2 start cloudsearch_s
on n3 start cloudsearch_s
on n2 wait for STARTED of cloudsearch_s
on n3 wait for STARTED of cloudsearch_s
#
# start API Portal Bundle
#
on n2 start apiportalbundle_s
on n3 start apiportalbundle_s
on n2 wait for STARTED of apiportalbundle_s
on n3 wait for STARTED of apiportalbundle_s
#
# finally, start loadbalancer
#
on n1 start loadbalancer_s
on n2 start loadbalancer_s
on n3 start loadbalancer_s
on n1 wait for STARTED of loadbalancer_s
on n2 wait for STARTED of loadbalancer_s
on n3 wait for STARTED of loadbalancer_s
```

2. Execute the script by running the following command from the command prompt

```
SAGInstallDir\API_Portal\server\acc>acc.bat
-n SAGInstallDir\API_Portal\server\nodelist.pt -c
SAGInstallDir\API_Portal\server\generated.apptypes.cfg -f
SAGInstallDir\API_Portal\server\startupScript.bat
```

   In the above command, replace the .bat files to .sh files, if you are using Linux OS. That is, acc.bat and startupScript.bat files must be replaced with acc.sh and startupScript.sh.

3. Ensure the HA setup is successfully running and all the runnables are started.

   In the default installation, access the User Management Component (UMC) at http://*machine name*/umc, the ARIS Document Storage (ADS) at http://*machine name*/ads, the collaboration component at http://*machine name*/collaboration, and the API Portal at http:/*machine name*.

   You can see that the elasticsearch cluster consists of three nodes, that the cluster name is apiportal and the master node is indicated by a solid star. Each index is split in 3 parts, called shards, which are replicated once and distributed over the three available nodes. If one node goes offline, the system can still fill up the complete index making it a fail-over system.

Managing Tenants

API Portal allows you to manage tenants for webMethods API Portal installation type. You can create or delete different tenants. You should be a user in the master tenant and have both Tenant administrator and User administrator privileges to create or delete a tenant.

Creating Tenants

You must be a user in the master tenant and have both the Tenant administrator and User administrator privileges to create a tenant. You can create any number of tenants.

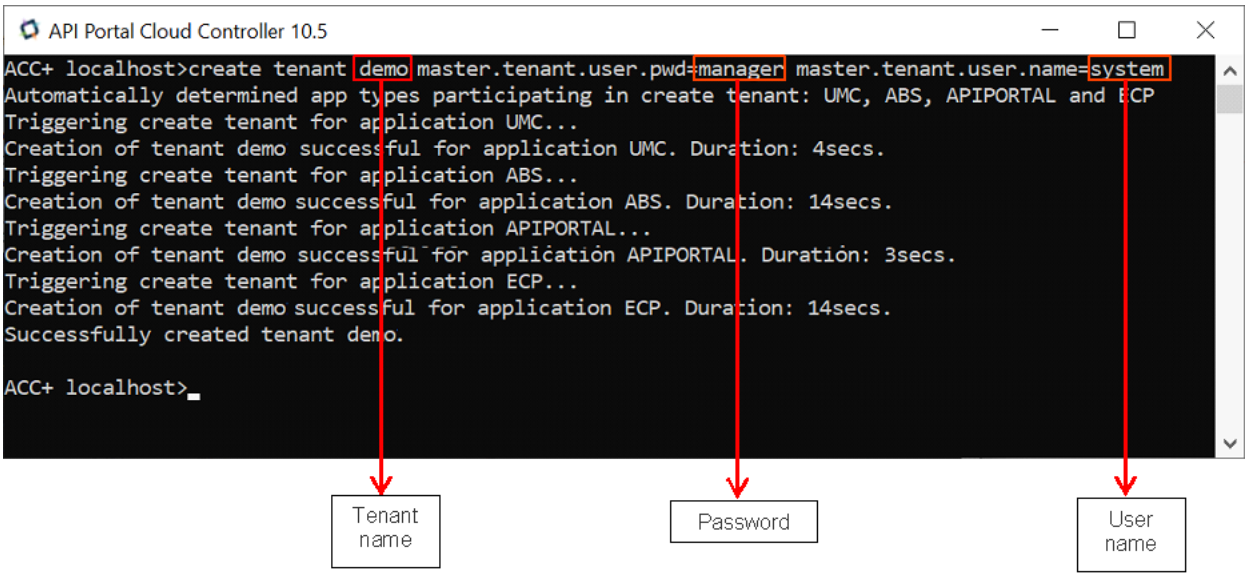
To create a tenant

1. Start the ACC.

2. Run the following command:

```
create tenant <tenant name> master.tenant.user.pwd=<password>
master.tenant.user.name=<username>
```

<tenant.name> is the name of the new tenant being created. If the master tenant user password has been modified, you can supply the credentials as part of create tenant command as shown below.



The tenant.user.pwd and tenant.superuser.pwd are the passwords that you want to set for the system and superuser accounts of the new tenant.

A new tenant is created with the given specifications. For example, we have created a new tenant called *demo*.

3. Run the following command to import license into the new tenant:

```
acc > invoke enhancement_importLicense on apiportalbundle_s tenant.name=
tenant.name local file enhancement.path="location to license file"
```

4. Run the following command to disable demo mode for the new tenant:

```
acc> invoke prepareTenant on apiportalbundle_s tenant.name=tenant.name
isDemo=false
```

Note:
If the tenant user name and password are changed, the default values can be overridden using parameters `tenant.user.name/tenant.user.pwd` as explained in step 1. The isDemo flag has to be set to true if you want to prepare a demo tenant similar to *sagtours*.

Note:
If any errors are encountered when creating or configuring tenant, check the logs located at <*SAG_InstallDir*> `\API_Portal\server\bin\work\work_apiportalbundle_s \base\logs\apiPortalTenantProvisioning.log`.

After you have created a tenant, you can configure a domain address for the tenant. For information on configuring domain name address for a tenant, see "Configuring Domain Address" on page 83.

Deleting Tenants

You should be a user in the master tenant and have both Tenant administrator and User administrator privileges to delete a tenant.

To delete a tenant

1. To delete a tenant execute the following command.

```
acc > delete tenant tenant.name
```

Note:
If the tenant user name and passwords are changed, you can supply the credentials as part of delete tenant command as follows:

```
acc > delete tenant tenant.name
       master.tenant.user.name=master.tenant.user.name
       master.tenant.user.pwd=master.tenant.user.pwd
```

Securing Internal Network Resources

When trying out an API that specifies localhost or any other internal network address, there is a possibility of getting the internal IP addresses as response. This can be prevented by executing a cross tenant property that disables intranet IP addresses being used for API tryout.

Enabling Internal Network Security

To run the cross tenant property:

1. Start the ACC.
2. Run the following command:

```
reconfigure apiportalbundle_s +JAVA-Dportal.tryout.enable.local.ips=false
```

The property is enabled and the access to the internal ports are secured.

Configuring the API Endpoint Validation

You can validate the certificate of the API Endpoints when trying APIs from the Try API page. If this validation is enabled, an error message is displayed during the following instances:

- The endpoint server's certificate is not issued by a trusted authority.
- The hostname of the endpoint server does not match the subject.

To enable API Endpoint validation

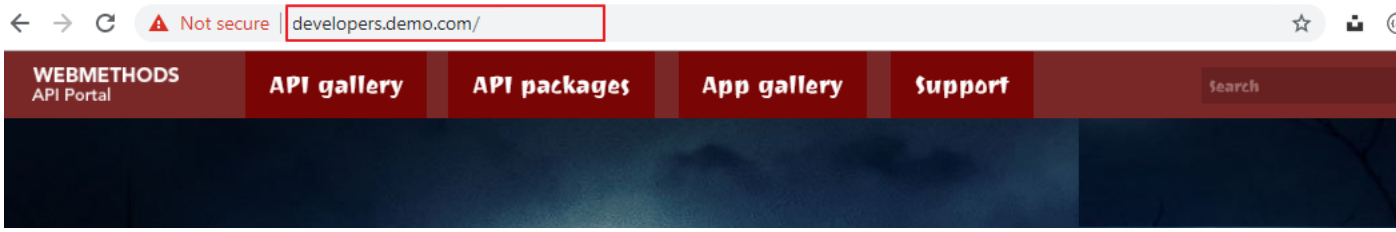
1. Start the ACC.
2. Run the following command:

```
reconfigure apiportalbundle_s +JAVA-Dportal.ssl.trust.all.enable=false
```

The certification validation is enabled.

Configuring Sub-domain Names for API Portal Tenants

API Portal allows you to configure required names to access your tenants. For example, if you have a tenant exclusive for the developers in your organization, you can configure the sub-domain name as follows: http://developers.demo.com.



The process of configuring sub-domain includes the following steps:

- Create a tenant. For information on creating tenants, see "Creating Tenants" on page 80.
- Configure sub-domain. For more information on configuring domain name, see "Configuring Domain Address" on page 83.

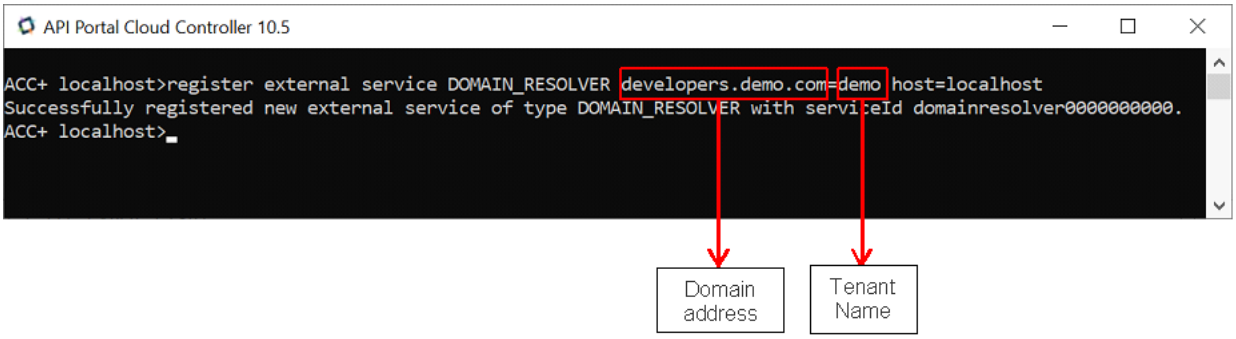
Configuring Domain Address

After a tenant is created, you can specify a sub-domain and map the tenant to the sub-domain. Users can access the tenant site from their browsers by providing the sub-domain name.

1. Start the ACC.
2. Run the following command with the tenant name and the corresponding domain address:

```
register external service DOMAIN_RESOLVER <domain_address>=<tenant_name>
host=localhost
```

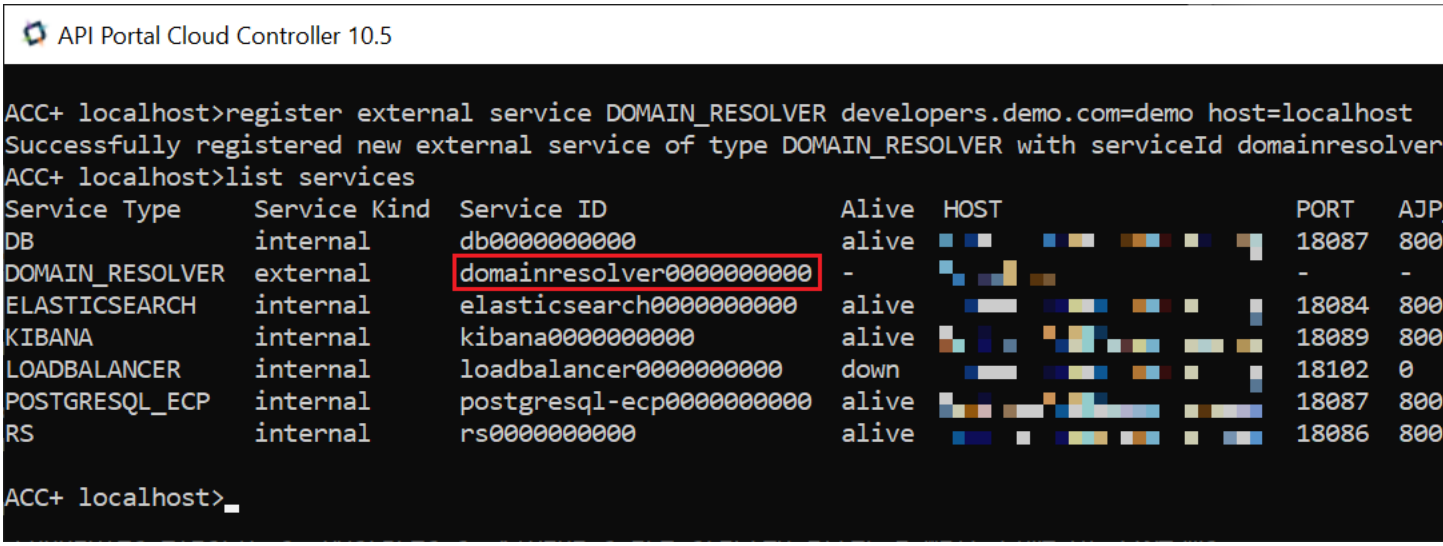
The figure depicts the domain address *developers.demo.com* being mapped to the tenant, *demo*.



3. Run the following command to retrieve the service ID of the **Domain_Resolver** runnable.

```
list services
```

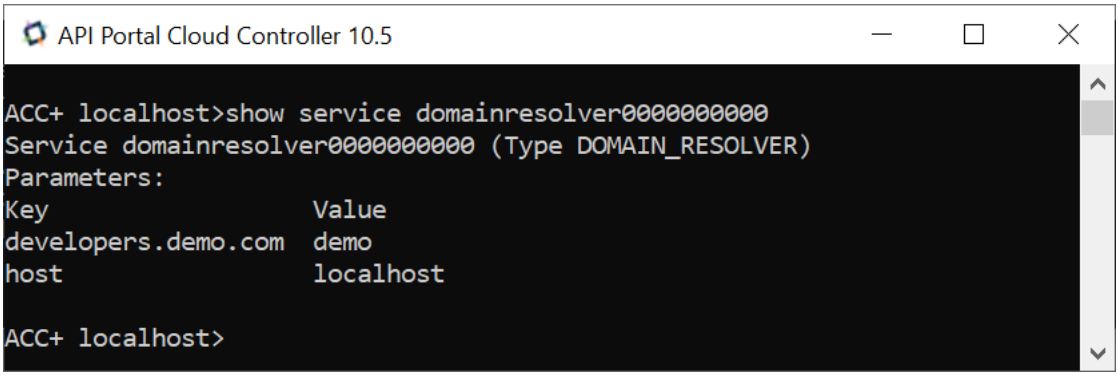
The list of services are displayed.



4. Copy the **Service ID** value of **Domain_Resolver** and run the following command:

```
Show service <service id>
```

The mapped domain address and tenant name are displayed.



5. Run the following command to disable the redirecting property of the **loadbalancer** runnable.

```
reconfigure loadbalancer_s +HTTPD.RewriteEngine=off
```

   The automatic redirection to the default tenant is disabled. If this step is not performed, the browser redirects users to the main tenant even when they enter the sub-domain name.

6. To validate the sub-domain configuration, perform one of the following:

   - For cloud hosting, the infrastructure team must create a DNS record for the configured sub-domain address.
   - For local installation, open the host file from the `C:\Windows\System32\drivers\etc\` and add the following entry:

     ```
     <local host> <domain address>
     ```

     For example, 127.0.0.1 developers.demo.com

7. Open a browser and provide the sub-domain name.

   You are redirected to the API Portal instance of the tenant you created.

3 Managing API Portal

Overview of Managing API Portal

Managing API Portal consists of starting and stopping API Portal and the API Portal Cloud Controller (ACC), verifying the status of all API Portal components, and opening the API Portal user interface in a browser to ensure that the portal looks and functions as intended.

What Happens When You Start API Portal?

When API Portal is installed from the Software AG Installer, the installer installs all the required API Portal components (also known as runnables).

On Windows, the API Portal Cloud Controller is installed as a Windows service.

You need to manually start API Portal following a machine restart.

Starting API Portal (Windows)

All API Portal components must be started before the API Portal user interface can be opened in a browser. If any of the components are not started, your browser displays an error. For example:

```
403 Forbidden You don't have permission to access / on this server.
```

To start API Portal on Windows

1. Start API Portal automatically or manually by doing one of the following:

   - **Automatic:** Start API Portal automatically using a Windows shortcut, as follows:

     Start > All Programs > Software AG > Start Servers > Start API Portal n.n

   - **Manual:** Start API Portal manually from the API Portal Cloud Controller (ACC), as follows:

     Start > All Programs > Software AG > Administration > API Portal Cloud Controller n.n

     In the command window, type `startall` to start all the components.

2. Verify the status of all API Portal components. For details, see "Verifying the Status of API Portal Components" on page 91.

Note:
You can configure API Portal in a way that all services start automatically. You can set the autostart mode with `ACC>set autostart.mode=all` to automatically start all the portal services once the Cloud Controller starts.

Starting API Portal (Linux/UNIX)

All API Portal components must be started before the API Portal user interface can be opened in a browser. If any of the components are not started, your browser displays an error. For example:

```
403 Forbidden You don't have permission to access / on this server.
```

To start API Portal on Linux/UNIX

1. Start the API Portal Cloud Controller (ACC) by running the *Software AG_directory* `/API_Portal/server/acc/acc.sh` script, specifying the following:

   - Machine on which the cloud agent is running (this will always be localhost) with the `-h` command line switch.

     Note:
     If the cloud agent is not running, you can start it by running the CloudAgentApp.sh script. Navigate to the *Software AG directory*`/API_Portal/server/bin` directory and run the command `./CloudAgentApp.sh start`.

   - Username (default: Clous) of the cloud agent user with the `-u` command line switch.
   - Password (default: g3h31m) of the cloud agent user with the `-pwd` command line switch.

     You can also omit the password. If you do so, the ACC prompts you for it.
   - Port information for the cloud agent with the `-p` command line switch.

   For example, to start the ACC installed in the directory SoftwareAG on localhost and using the default username and password, use the command:

   ```
   SoftwareAG/API_Portal/server/acc/acc.sh -h localhost -u Clous -pwd g3h31m
   -p 18011
   ```

2. At the ACC command prompt, type `startall` to start all components..

   Note:
   Ensure that you have installed the **fontconfig** library for seamless starting of all components. If it is not installed already, install the same by running the following command:

   ```
   sudo yum install fontconfig
   ```

3. Verify the status of all API Portal components. For details, see "Verifying the Status of API Portal Components" on page 91.

Note:
You can configure API Portal in a way that all services start automatically. You can set the autostart mode with `ACC>set autostart.mode=all` to automatically start all the portal services once the Cloud Controller starts.

Stopping API Portal (Windows)

To stop API Portal on Windows

1. Stop API Portal manually or automatically by doing one of the following:

- **Automatic:** Stop API Portal automatically using a Windows shortcut, as follows:

  Start > All Programs > Software AG > Start Servers > Stop API Portal n.n

- **Manual:** Stop API Portal manually from the API Portal Cloud Controller (ACC), as follows:

  Start > All Programs > Software AG > Administration > API Portal Cloud Controller n.n

  In the command window, type `stopall`.

2. Verify the status of all API Portal components. For details, see "Verifying the Status of API Portal Components" on page 91.

Stopping API Portal (Linux/UNIX)

To stop API Portal on Linux/UNIX

1. Start the API Portal Cloud Controller (ACC) by running the *Software AG_directory* /API_Portal/server/acc/acc.sh script, specifying the following:

   - Machine on which the cloud agent is running (this will always be localhost) with the `-h` command line switch.

     Note:
     If the cloud agent is not running, you can start it by running the CloudAgentApp.sh script. Navigate to the <*Software AG directory*>`/API_Portal/server/bin` directory and run the command `./CloudAgentApp.sh start`.

   - Username (default: Clous) of the cloud agent user with the `-u` command line switch
   - Password (default: g3h31m) of the cloud agent user with the `-pwd` command line switch

     You can also omit the password. If you do so, the ACC prompts you for it.

   - Port information for the cloud agent with the `-p` command line switch.

   For example, to start the ACC installed in the directory Software_AG on localhost and using the default username and password, use the command:

   ```
   Software_AG/API_Portal/server/acc/acc.sh -h localhost -u Clous -pwd g3h31m
   -p 18011
   ```

2. At the ACC command prompt, type `stopall`.

   Note:
   Ensure that you have installed the **fontconfig** library for seamless starting of all components. If it is not installed already, install the same by running the following command:

   ```
   sudo yum install fontconfig
   ```

3. Verify the status of all API Portal components. For details, see "Verifying the Status of API Portal Components" on page 91.

API Portal Components

API Portal includes the following components (runnables). All must be in the STARTED state for the API Portal user interface to come up in the browser.

Instance ID (runnable)	Description
zoo_s	Service registry
postgres_s	PostgreSQL database
cloudsearch_s	Intelligent API search capabilities for API Portal.
elastic_s	Search storage engine with search capabilities
kibana_s	Data visualization and analytics engine.
apiportalbundle_s	API Portal business logic.
loadbalancer_s	Load balancer to distribute the request load across servers.

Verifying the Status of API Portal Components

Use the ACC to manually manage API Portal components. To monitor the status of all API Portal components (runnables), use the `list` command.

In the following example, the response from the `list` command shows the component apiportalbundle_s with a status of STARTED.

```
ACC+ localhost>list

On node localhost 7 runnables are installed.

zoo_s          : STARTED (com.aris.runnables.zookeeper-run-prod-99.0.0.0-a-
_trunk-SNAPSHOT)

postgres_s     : STARTED (com.aris.runnables.PostgreSQL-run-prod-99.0.0.0-a-
_trunk-SNAPSHOT)

cloudsearch_s  : STARTED (com.aris.cip.y-cloudsearch-run-prod-99.0.0.0-a-
_trunk-SNAPSHOT)

elastic_s      : STARTED (com.aris.runnables.elasticsearch-run-prod-99.0.0.0-a-
Trunk-SNAPSHOT)
kibana_s       : STARTED (com.aris.runnables.kibana-run-prod-99.0.0.0-a-
Trunk-SNAPSHOT)

apiportalbundle_s : STARTED (com.aris.apiportalbundle.y-apiportalbundle-run
-prod-99.0.0.0-a-_trunk-SNAPSHOT)

loadbalancer_s : STARTED (com.aris.runnables.httpd.httpd-run-prod-99.0.0.0-a-
```

```
_trunk-SNAPSHOT)
```

For more details about the API Portal components and their status in the ACC, see " API Portal Components" on page 91 and "Understanding API Portal Component Status in ACC" on page 92.

Understanding API Portal Component Status in ACC

When using the `list` command in the ACC, the command response lists components by their component name (runnable instance ID), followed by the state of the component. Possible states are as follows:

State	Meaning
UNKNOWN	The component state is not yet known. This state is normally only shown directly after the ACC service is (re-)started.
STOPPED	The component is currently not running.
STARTING	The component is starting, but this process is not yet complete.
STARTED	The component is running.
STOPPING	The component is stopping, but this process is not yet complete.
DOWN	The component has crashed. The agent attempts to automatically restart the component momentarily.
FAILED	The component has crashed. The agent has given up trying to restart the component (or automatic restarting has been disabled).

If a component does not start properly, look at the logs of the component. The logs are available at *Software AG_directory* `\API_Portal\server\bin\work\work_`*component_name*`\base\logs`. If you need additional help to determine why components are not starting, contact Software AG Global Support.

Starting and Stopping API Portal Components

API Portal components can be started and stopped independently, but most components do not work on their own.

Start the ACC and type these commands at the prompt:

Command	Description and Notes
`startall`	Starts all API Portal components in the correct order. To monitor the progress, use the `list` command. If successful, the system responds: Successfully started all not yet running runnables on node localhost

Command	Description and Notes
`start instanceId`	Starts the specified API Portal component. For example, the command to start the apiportalbundle_s component is: `start apiportalbundle_s` If successful, the system responds: Successfully started runnable apiportalbundle_s on node localhost If issues arise, ACC returns additional information. For example: Could not start runnable apiportalbundle_s on node localhost: Ant stop exited with 1
`stopall`	Stops all API Portal components. To monitor the progress, use the `list` command. If successful, the system responds: `Successfully stopped all running runnables on node localhost.`

`stop instanceId`	Stops the specified API Portal component.
	For example, the command to stop the apiportalbundle_s component is:
	`stop apiportalbundle_s`
	If successful, the system responds:
	`Successfully stopped runnable apiportalbundle_s on node localhost`
	If issues arise, ACC returns additional information. For example:
	`Could not stop runnable apiportalbundle_s on node localhost: Ant stop exited with 1`

Opening the API Portal User Interface in a Browser

To open the API Portal user interface, open your browser and point it to the port on the host machine where the API Portal instance is running. By default, API Portal runs on port 18101.

All API Portal components must be started to open the API Portal user interface. If any of the components are not started, the browser displays an error. For example:

```
403 Forbidden You don't have permission to access / on this server.
```

To open the API Portal user interface

1. Start your browser, and then point it to the host and port where API Portal is running. For example:

   - If API Portal is running on the default port on the same machine where you are running the API Portal components, you would enter:

     ```
     http://localhost:18101
     ```

   - If the API Portal components are running on port 4040 on a machine called QUICKSILVER, you would enter:

     ```
     http://QUICKSILVER:4040
     ```

2. In the API Portal login screen, log on with your user name and password.

   If you are logging in to perform administration tasks, log on with your API Portal Administrator credentials. The default values are:

   - User Name: system
   - Password: manager

Changing the Password

Change your password in the user profile section after your first login or after a password reset by the administrator.

To change the password

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **<user name>**. Your profile is displayed.
3. Click **Change password**.
4. In the Change Password section, type the current password.
5. Type a new password.
6. Re-type the new password to confirm.
7. Click **Save Changes**.

The password is changed. The user receives a notification by e-mail.

Editing your Profile

You can modify your profile account settings in the My Profile section.

To edit your profile

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **<user name>**. Your profile is displayed.
3. Click the section you want to edit, for example **Account settings**.
4. Change the setting.
5. Click **Save Changes**.

You changed your profile data.

Configuring Display Settings

Change the display settings to view the display in a language of your choice.

To configure display settings

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **<user name>**. Your profile is displayed.
3. Click **Display settings**.
4. In the Display settings section, select a required language for display from the drop-down list for **Language**.

   The available languages are: **German (Germany)**, **English (United States)**, **Spanish (Spain)**, **French (France)**, **Italian (Italy)**, **Japanese**, **Dutch (Netherlands)**, **Polish**, **Portuguese (Portugal)**, **Russian (Russia)**.
5. Click **Save Changes**.

Scheduling Reports for Application Usage

You can schedule reports that provide periodic alerts on the usage of applications you own. The application usage metrics are collected and mailed to the configured email address.

To schedule reports

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click user name.

This displays your profile.

3. Click **Reports**.

4. In the Reports settings section, select a value for the frequency at which the report is generated and sent to the configured email. Select one of the following:

   - **Daily**: Specifies that the reports are generated daily and provides the usage of applications on daily basis. Reports are scheduled @12 PM every day.
   - **Weekly**: Specifies that the reports are generated every week and provides the usage of applications for the last seven days. Reports are scheduled @12 PM every Friday.
   - **Monthly**: Specifies that the reports are generated every month and provides the usage of applications for a month. Reports are scheduled @12 PM on the first day of the month.

5. Click **Save Changes**.

   The reports are now scheduled and emailed to the configured email as per the frequency set. The report sent out has two sections.

   - **Summary**: Consists a summary in the form of a list of applications that the user owns and corresponding usage metric.
   - **Details**: Consists detailed information for each application, the API associated with the application, Resources used, and usage count.

   Note:
   As an API Portal administrator, you can configure the email template, which is used for sending the Application Usage report, in the Configuration settings page.

Managing Teams

You can create teams of required users to share an application exclusively with users who are part of a team. You can create, edit, or delete teams from the Manage Teams page.

Creating Teams

To create a team

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **<user name>**. Your profile is displayed.

3. Click **My teams**.

   The Manage Teams page is displayed. This page displays the list of existing teams.

4. Click **Create**.

   The Create teams and associate users page is displayed.

5. Type the name of the team and description in the **Name** and **Description** fields respectively.

6. Click **Users**.

   The Select User(s) page is displayed.

7. Select the required user(s) from the Available Users section. To search for a user, type the user name in the text box in the section.

8. Click **Add**. To add all users displayed in the Available Users section, click **Add All**.

   The selected users are displayed in the Associated users section.

9. Click **Ok**.

10. Click **Save**.

The team details are saved.

Editing Teams

To edit the details of a team

1. Click  in the right top corner of the API Portal window to display the menu options.

2. Click **<user name>**. Your profile is displayed.

3. Click **My teams**.

   The Manage Teams page is displayed. This page displays the list of existing teams.

4. Click the edit icon next to team that you want to edit.

   A details of the selected team are displayed.

5. Make the required changes and click **Yes**.

The edits are saved.

Deleting Teams

To delete a team

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **<user name>**. Your profile is displayed.
3. Click **My teams**.

   The Manage Teams page is displayed. This page displays the list of existing teams.
4. Click the delete icon next to team that you want to delete.

   A confirmation message is displayed.
5. Click **Yes**.

The team is deleted.

Note:
If you delete a team, the members who were part of the team will no more have access to the application(s) shared with the team.

Managing Notifications

You can view notifications, take the required action and configure the notification settings as required.

To change the password

1. Click in the title bar.

   The notifications are listed in a drop down box.
2. You can manage the notifications as follows:
   - Click on any notification.

     It navigates you to the appropriate page. For example, it navigates to the API Details page when there is a notification about API update.
   - Click **Show all** to display all the notifications. The **Show all** option is displayed if there are more than five notifications.
     - Select a notification and click **Mark as read** to mark the notification as read.
     - Select a notification and select **Delete** to delete the notification.
     - Select **Notification settings** to configure notification settings.
   - Click on to configure notification settings.

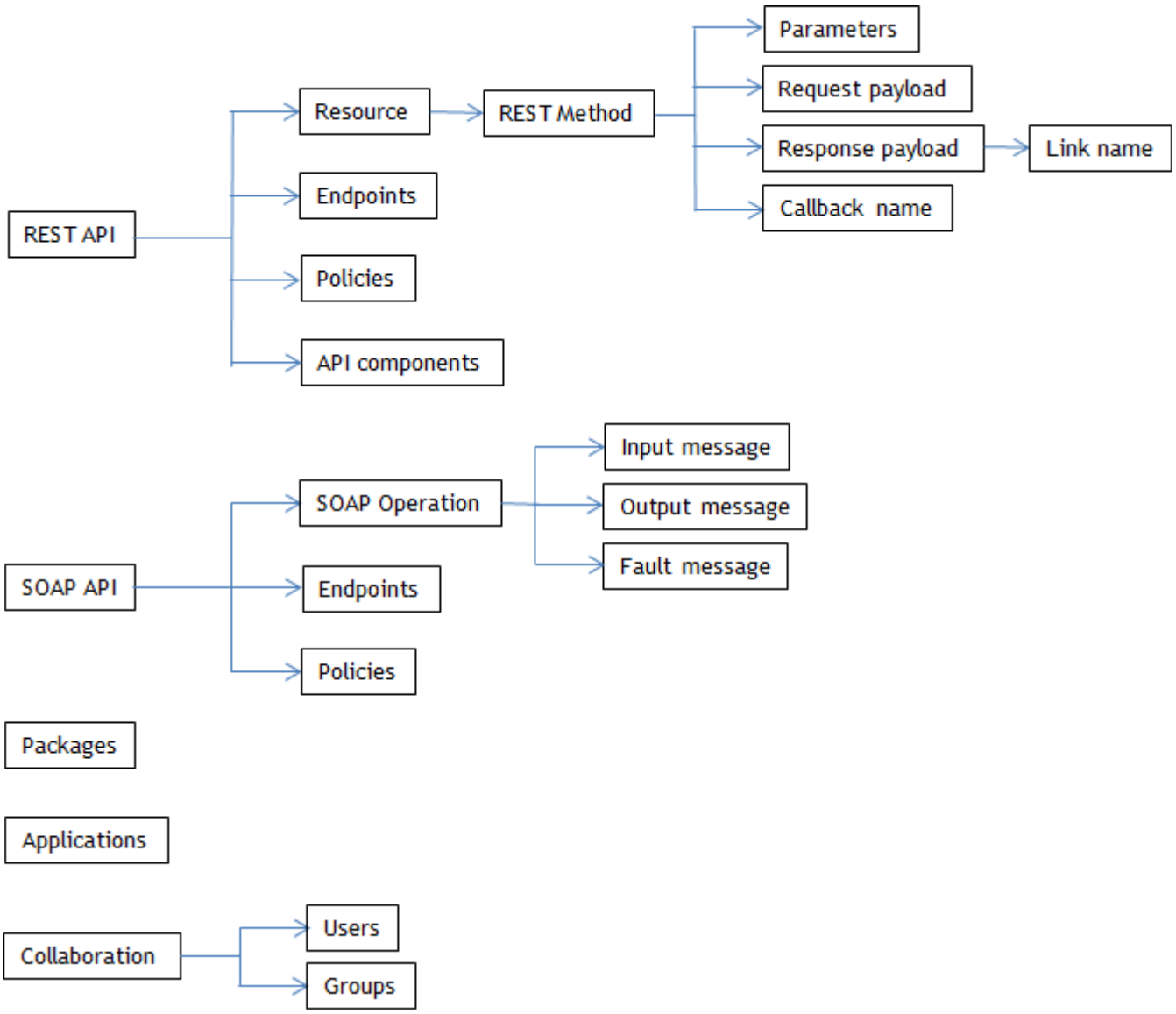
Select the required options, in the Notification settings view, to configure the notification setting accordingly.

- Click **Mark all as read** to mark all the notification messages as read.
- You can do a bulk delete or read by selecting the checkbox available to the left of each notification.

Searching in API Portal

The search feature in API Portal is a type-ahead search; a simple and easy to use search facility where you can type the text of interest to search. You can search for all items that contain one or more specified keywords (that is, text strings) in the item's properties. Some of the properties are name, description, version, key, value, tags, and so on.

This figure depicts the various items searchable in API Portal and the table lists the attributes that are searchable for each item type.



This table lists the attributes that are searchable for API-specific data:

Item	Attributes
REST API	■ Name ■ Short description ■ Description ■ Version ■ API group ■ Tags ■ Business term

Item	Attributes
	■ Maturity state ■ Other attributes ■ API component name
SOAP API	■ Name ■ Short description ■ Description ■ Version ■ API group ■ Tags ■ Business term ■ Maturity state ■ Other attributes
OData API	■ Name ■ Short description ■ Description ■ Version ■ API group ■ Tags ■ Business term ■ Maturity state
REST Resource	■ Name ■ Path ■ Description ■ Tags ■ Callback name
REST Method	■ Name ■ Description ■ HTTP Method

Item	Attributes
	■ Tags
Parameter	■ Name ■ Description
Request payload	■ Schema ■ Sample request
Response payload	■ Schema ■ Sample response ■ Link name
SOAP Operation	■ Name ■ Description ■ Tags
SOAP message	■ Input message ■ Output message ■ Fault message
Endpoints	■ Base URL
Policies	■ Name ■ Description
Package	■ Name ■ Description
Applications	■ Name ■ Description
Entity	■ Name ■ Type ■ Container ■ Entity sets
Complex types	■ Name ■ Type
Functions	■ Name

Item	Attributes
Actions	■ Name
Singleton	■ Name ■ Type
Properties	■ Name ■ Type

This table lists the attributes that are searchable for Collaboration data:

Item	Attribute
Users	■ First name ■ Last name ■ User name
Groups	■ Name ■ Description

To search for an object, type a string in the search box in the title navigation bar. A list of search result is displayed directly below the Search box. The number of matches found are displayed in two sections, APIs and Collaboration. A minimum of five search results are displayed in each category. The total number of matches across categories is listed at the bottom of the search box. If there are no results as per the search string typed, a message displays saying so.

If you find what you are searching for in the search result box, click on it to view the details. You are navigated to the specific page that displays more information. For example, if you are searching for a parameter in a REST method and click the displayed result, you are navigated to the parameters section in the corresponding API details page.

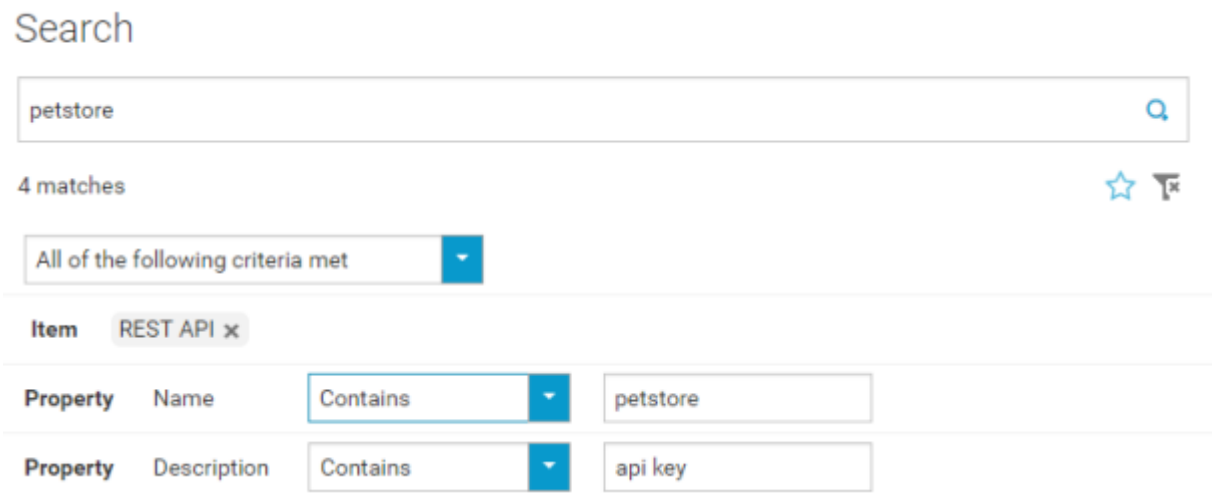
If you want to see all the search results click **Show all** in the search result box. The Advanced search page is displayed. This is a dedicated page that displays extensive search results. The search results are categorised in two tabs, APIs and Collaboration. You can select the respective tab to see search results pertaining to the category. A search box is available in the Advanced search page; you can search for a particular string by typing the string in the search box and only results that contain the string are displayed.

You can filter the search results in the two tabs as follows:

- **APIs tab**: You can filter based on item types and include a subordinate filter of properties for each item type.

  For example, type a search string `swagger` in the search box on the title navigation bar. Click **Show all** to view all the search results that match the search string. This displays all the results in the advanced search page categorized under APIs and Collaboration tabs. In the APIs tab,

you can now use a filter **REST API** in the list of filters for items; this filters the list to display only entries that have the specified item. You can further filter the list depending on the properties for the selected item; for example, select the properties **Name** and **Description**. You can provide one of the filtering criteria, **All of the following criteria met** or **One of the following criteria met**. Provide additional details for a string in the specified properties; **Name contains** `petstore` and **Description contains** `api key`. The figure displays the search criteria applied in this example.



The search result list now displays only the search result items that are specified as per the filter.

- **Collaboration tab**: You can filter based on Users and Groups. For example, if you select the search result that describes the search item to be found for a User, it navigates to that particular user's page to display the occurrence.

You can clear the filters in each of the tabs by clicking and add new filters by clicking the required item or property.

You can save the filters as favourites by clicking and delete the saved favourites by clicking .

Note:
When a backup taken from previous versions is restored in 10.1, the applications and its related properties are displayed as (Untitled). This is resolved by republishing the corresponding APIs to API Portal.

4 Managing Users

Overview of Managing Users

API Portal users are managed through API Portal User Management Console (UMC). The roles and permissions assigned to the users signify who can publish the APIs to API Portal and access API Portal.

Users identify individuals that are known to API Portal, CentraSite or API Gateway. The roles and permissions that are assigned to users specify which operations they can perform and what sections they can access.

For more information about CentraSite users, groups, roles, and permissions, see *CentraSite Administrator's Guide*.

For more information about API Gateway users, roles, and permissions, see *webMethods API Gateway User's Guide*.

User Roles and Groups in API Portal

API Portal provides predefined roles that you can assign to users and groups defined in an organization. You can also create custom roles as needed. Users or groups who have roles receive all permissions associated with the roles.

The following is a list of the roles and function privileges in API Portal that apply to API users and administration. For complete information about the predefined roles and creating custom roles in API Portal, see the API Portal User Management help, available from http://*API_Portal_host*:*port*/umc/help/en/handling/index.htm.

User Roles and Groups	Description
API Administrator	Users with this role can start and stop API Portal, manage API Portal users, customize the API Portal user interface to reflect the organization's own branding and look and feel, and switch configuration sets to customize views in API Portal. API Administrators can create and remove private communities and can also manage all communities. API Administrators can add and remove users from a community and define community administrators or revoke the community administrator role from a user.
API Provider	An API provider is allowed to publish APIs to API Portal. These users are registered in CentraSite, API Gateway and APIs are published to API Portal.
API Consumer	An API consumer is allowed to browse the portal, request API access tokens, and test (evaluate) available APIs.
API User Registration Approvers	This is a group of users who are notified when there is a user registration request for a new user. This group of users are assigned permissions to approve or reject any user registration requests.

User Roles and Groups	Description
API Consumption Approvers	This is a group of users who are notified when there is a request for API consumption. This group of users are assigned permissions to approve any API consumption request.
Public Community	This is a group that an on-boarded user is added to, by default.

In addition to these roles, technical users exist to facilitate communication between systems and applications to ensure that credentials stay the same. A technical user is not associated with a specific user. Rather, a technical user represents a set of credentials and authorizations that is authenticated against an internal list of users, and not with an external set of authentications (for example, Active Directory or LDAP). API Portal administrators create technical users in API Portal, and CentraSite administrators specify the technical user credentials when they register an API Portal instance in CentraSite. Guest users are anonymous users who can browse and test the APIs available in API Portal. When a guest user decides to use an API, the user must register and request an access token.

Note:
As a best practice, Software AG recommends using a technical user in CentraSite and API Gateway to publish APIs to API Portal.

Importing LDAP Users and User Groups into User Management Console

You can import users and user groups from the LDAP system.

1. Log on to UMC as an API Portal Administrator.
2. Click User management.

   The list of users is displayed
3. Click  **Additional functions**
4. Click  **Start LDAP import**.

   The button is active only if an LDAP system is configured on the server.
5. Select whether you want to import only users or user groups and associated users.
6. Select if you want to use the default filter or create a custom one.
7. Click **Preview** to check how many users or user groups are imported.

   The number is displayed, as well as up to 100 elements to be imported in alphabetical order.

8. Click **Start import**.

   The users are transferred from the LDAP system according to the selected options. The imported users can now log on to API Portal.

Synchronizing LDAP Users or User Groups with User Management Console

You can synchronize LDAP users and user groups with UMC.

1. Log on to UMC as an API Portal Administrator.
2. Click **Configuration**.
3. Click the arrow next to **LDAP**.

   This lists various configuration options available.
4. Click **General settings > Advanced settings**.
5. Click  **Edit**.
6. Select **Import user at login** and **Import user groups when synchronizing**.
7. Click **Save**.

   The user synchronizes to UMC on the user's first login to API Portal.

Assigning Privileges to User Groups

The predefined user groups are assigned with the privileges based on their corresponding roles. When you create a user group, you must assign the necessary privileges to the group for the users to perform required transactions.

To assign privileges

1. Log on to UMC as an API Portal Administrator.
2. Click **User management**.

   The list of users is displayed.
3. Click **User groups**.

   The list of user groups is displayed.

4. Click the user group that you want to assign privileges.

   The details of the selected user group are displayed.

5. Click **Privileges**.

   The list of privileges are displayed.

6. Select the **Granted** check box next to the privilege that you want to assign.

   You can refer the **Description** of a privilege to learn its purpose.

7. Click **Licensed privileges**.

8. Select the **Granted** check box next to the **webMethods API Portal Viewer** privilege to view the group **Approval Settings** section in User Registration Setting screen of API Portal.

   The selected privileges are assigned to the user group.

Password Policy for API Portal Users

A password policy is a set of rules designed to enhance security by encouraging users to employ strong passwords and use them properly. This is configured through the User Management Console (UMC). The password policy compliance is checked in the following scenarios:

- **New user signup**: When a new user signs up, the password provided must be compliant to the password policy.
- **Password update in user profile**: When a user updates the password, the new password provided should be compliant with the password policy.
- **New user invite from communities**: When a user receives an invite from communities, a random password is generated and sent to the user. This password should be compliant to the password policy.

The following parameters can be configured in the **Configuration > Password policy** section under various categories in UMC. Alternatively, you can also configure this in the **Configuration > All** section by setting the parameters mentioned. The table lists the parameters, description and their corresponding properties.

Parameter	Description
Minimum length under **Password policy > General**.	Specifies the minimum length of a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.length.min`

Parameter	Description
Maximum length under **Password policy > General**.	Specifies the maximum length of a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.length.max`
Minimum number of lowercase letters under **Password policy > General**.	Specifies the minimum number of lowercase alphabets in a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.characters.lowercase.min`
Allow uppercase letters under **Password policy > General**.	Specifies whether the uppercase alphabets are allowed in a password. Valid input: true, false **Property:** `com.aris.umc.password.characters.uppercase.allowed`
Minimum number of uppercase letters under **Password policy > General**.	Specifies the minimum number of uppercase alphabets in a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.characters.uppercase.min`
Allow numbers under **Password policy > General**.	Specifies whether numbers are allowed in a password. Valid input: true, false **Property:** `com.aris.umc.password.characters.numeric.allowed`
Minimum number of numbers under **Password policy > General**.	Specifies the minimum number of numerals that must be contained in a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.characters.numeric.min`

Parameter	Description
Allow special characters under **Password policy > General**.	Specifies whether special characters are allowed in a password. Valid input: true, false **Property:** `com.aris.umc.password.characters.special.allowed`
Minimum number of special characters under **Password policy > General**.	Specifies the minimum number of special characters in a password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.characters.special.min`
Special characters under **Password policy > General**.	Specifies which characters are special characters. Valid input: String **Property:** `com.aris.umc.password.characters.special.set`
Activate expiring passwords under **Password policy > Expiring passwords**.	Specifies whether passwords are set to be valid only for a specific amount of time. This is defined for a single tenant. Once the password has expired, the user is directed to a website enabling the password to be changed. Thereafter, the user is redirected to the application. Valid input: true, false **Property:** `com.aris.umc.password.expiry.active`
Password lifetime under **Password policy > Expiring passwords**.	Specifies the period of time after which a password expires. Valid input: Integer > 0 **Property:** `com.aris.umc.password.expiry.days`
Force change after reset under **Password policy > Advanced settings**.	Specifies whether a user must change the password if it was reset (and sent through an e-mail). Valid input: true, false **Property:** `com.aris.umc.password.change.forceAfterReset`

Parameter	Description
Force different password under **Password policy > Advanced settings**.	Specifies whether the new password must differ from the old one. Valid input: true, false **Property:** `com.aris.umc.password.change.forceDifference`
Force change before first login under **Password policy > Advanced settings**.	Specifies whether a user must change the password upon first login. Valid input: true, false **Property:** `com.aris.umc.password.change.forceOnFirstLogin`
Activate reset confirmation under **Password policy > Advanced settings**.	Specifies whether a user must confirm a password reset. Valid input: true, false **Property:** `com.aris.umc.password.reset.confirmation.active`
Link lifetime under **Password policy > Advanced settings**.	Specifies the time in seconds during which a user can click the link sent by e-mail in order to confirm the password. Valid input: Integer > 0 **Property:** `com.aris.umc.password.reset.confirmation.ttl`

Configuring password policy for API Portal Users

You can configure the password policy to enhance security by encouraging users to employ strong passwords. You should have API Portal Administrator privileges to configure the password policy.

1. Log on to UMC as an API Portal Administrator.
2. Click **Configuration**.
3. Click **Password policy**.
4. Click the required category.

   Available categories are **General**, **Expiring passwords**, and **Advanced settings**.

5. Click to edit the parameters.
6. Provide values for various parameters as required.
7. Click **Save**.

   This saves the configuration applied for the password policy.

Configuring LDAP Servers

LDAP enables information from a distributed, location-independent and hierarchical database in a network to be queried and modified. You can use multiple LDAP servers with API Portal. Any existing LDAP data needs to be deleted manually if you wish not to have the data. You can configure a single or multiple LDAP servers.

You have the Administrator Role to perform this task.

To configure LDAP servers

1. Log on to UMC as an Administrator.
2. Click **Configuration**.
3. Click the arrow next to **LDAP**.

   This lists various configuration options available.
4. Click **General settings**.
5. Click **Edit**.
6. Select **Activate LDAP**.

   Select this option if you want to configure just one LDAP server.
7. Select **Activate multiple LDAP integration**.

   Select this option if you want to configure multiple LDAP servers.
8. Click **Save**.
9. You can add LDAP servers in one of the following ways:

   - You can add individual LDAP server and repeat the following steps to add multiple LDAP servers.

     1. Click **Add**.

The Add LDAP server dialog opens.

2. Provide the required information:
   - ID of LDAP server.
   - Display name of the LDAP server.
   - LDAP server URL.
   - LDAP server fallback URL
   - User name of the user who has access to the LDAP content.
   - Password of this user.
   - Specify whether or not SSL should be used and in which mode.
   - Specify whether the host names and certificates should be verified.
   - Specify the connection timeout.
   - Specify the read timeout.
   - Click **Save**.

- You can configure LDAP servers by importing a configuration file. The configuration file can have configuration details for one or more LDAP servers.

  1. Click  to import a configuration file.
  2. Click **Select file**.
  3. Select the required file.
  4. Click **Open**.
  5. Click **Upload**.

     The configuration file is uploaded and the configuration applied.

     Note:
     The uploaded configuration file overwrites the existing LDAP configuration.

10. Click the arrow next to **LDAP**.

    It displays the LDAP servers added below the **General settings** option. Click on individual LDAP servers to view the details of the LDAP server and test the connection by clicking  .

    You can download the configured LDAP servers information as a configuration file by clicking  the export icon.

When there are multi LDAPs configured, a user has to login with *ldap_id**user_id* during login.

Creating Truststore for LDAP

After you configure LDAP settings, you need to create a Truststore file and import the certificate generated in the Truststore file.

Pre-requisites

- The required Truststore file.
- Ensure the KeyStore Explorer application is installed and available with the JRE

Note:
API Portal does not consider the truststore file uploaded for LDAP server from UMC.

1. Open the JVM truststore of API Portal from the following location:
   `<install_dir>\API_Portal\server\jre\lib\security\cacerts`.
2. Click **Tools** and select **Import Trusted Certificate**.
3. Select the truststore file and click **Import**.
4. Save the cacerts file and restart the apiportalbundle_m runnable.

Enabling Multi-factor Authentication

API Portal provides multi-factor authentication (MFA) that requires the use of two or more authentication factors to verify a user's identity for a login. Authentication factors can be classified into knowledge factors (what the user knows, for example, password), possession factors (what the user has, for example, security token) and inherence factors (what the user is, for example, biometric verification). The authentication mechanism validates each factor thus adding another layer of security during a user log on.

API Portal uses a combination of username, password, and a one-time password (OTP) as authentication factors to verify the user's identity. The user receives the OTP in one of the following ways:

- Through an email: a user can request a new OTP which is sent to the user through email.
- As a secret token in an email: a user can use the secret token and generate an OTP using an external client, such as Google Authenticator.

You can enable this feature in the API Portal user management console.

Any user when on-boarded onto API Portal receives a secret token through an email, when MFA is enabled. The user can use this secret token to generate an OTP, using an external client like Google Authenticator, which in turn is used to log onto API Portal.

To enable multi-factor authentication

1. Log on to UMC as an Administrator.
2. Click **Configuration**.
3. Click **Security > Multi-factor authentication** in the left navigation pane.
4. Click  .
5. Select **Use multi-factor authentication** to enable it.

   Alternatively, you can also set the configuration property `com.aris.umc.authentication.multiFactor.active` as **true** under Configuration > All section. You can provide a value for **Clock skew intervals** or use the configuration property `com.aris.umc.authentication.multiFactor.clockSkew` to set the interval for which the generated OTP is valid. Each interval is 30s.

   Note:
   When you enable MFA and if you want few users to be excluded from MFA, you can add the multiple users separated with comma, under the **Excluded users**. By default all the system users are included in this list.

6. To generate and send out a secret token to users who were onboard before enabling multi-factor authentication, do the following:

   a. Click **Configuration**.

   b. Click **All** in the left navigation pane.

   c. Ensure that the property `com.aris.umc.notification.otpSecretChanged.enabled` is set to **true**.

   d. Click **User management** in the title navigation bar.

   e. Click the required user.

   f. Click **Generate token secret**.

      A new token is generated and sent to the respective user.

   Note:
   The user receives a mail with the token secret which can be used to generate an OTP to log on to API Portal.

   These steps must be performed for every user who was onboarded before MFA was enabled.

Configuring SAML 2.0 for a Consumer User

If an API Consumer needs to login to API Portal with SAML authentication, the user needs to have API Consumer role even before the first login. API Consumer role can be assigned by API Administrator using UMC.

1. Log on to UMC as an Administrator.
2. Click **Configuration**.
3. Click **SAML** in the left navigation pane.
4. In the General section provide the following information:
   a. **Identity provider ID**: The ID that was used while configuring Single sign-on.
   b. **Service provider ID**: The ID that was used while configuring Single sign-on.
   c. **Single sign-on URL**: The SingleSignOnService location POST url from SSO configuration.

      Alternatively, you can change the property key in the Configuration section as `com.aris.umc.saml.identity.provider.sso.url`
5. Click **Keystore**.
6. Provide the required information for **Keystore** value, **Alias**, **Password**, and **Type**.
7. Click  to upload the keystore that was created while configuring Single Sign-on.
8. Click **Truststore**.
9. Provide the required information for **Truststore** value, **Alias**, **Password**, and **Type**.
10. Click  to upload the truststore that was created while configuring Single Sign-on.
11. Activate SAML by selecting **Use SAML** in the General section.

    Alternatively you can change the property `com.aris.umc.saml.active` to value true to activate SAML.

5 Managing API Providers

Manage API Providers

An API Provider has the privileges to enable an API Portal Administrator or Provider to manage APIs, and configure notification types that are used for API-related events. Notification configuration is required to notify the API Provider regarding any event for an associated API, like a token request. API Portal supports two types of notifications:

- **Email**. Email alias of a user group in UMC.
- **HTTP Callback**. The URL to which the notification event data is sent. The format of data is available in the API Portal Configurations page.

There are two types of API providers in API Portal:

- **Default Provider**. Any API that is imported and not associated with any other API provider is automatically associated with the Default Provider. When an imported API is associated with a new API provider, it gets disassociated from the Default Provider. The Default Provider has only the e-mail notification configured. By default, you will find the Default Provider in the Manage Providers page.

  Only the details in the email alias group is editable.

- **External Provider**. Any other provider falls under this category.

  Except CentraSite and API Gateway, the details of all other external providers are editable.

The details you can view and the actions you can perform in this page are listed in the following table:

Field	Description
Name	Name of the API Provider.
Description	Description of the API Provider.
APIs	The number of APIs associated with the API Provider. Hovering over the number displays the APIs that are associated.
Action	Actions that can be performed on the API Provider like Edit and Delete.
Create	Click to create an API Provider.

Creating an API Provider

You must have the privileges of an API Portal Administrator or an API Provider to create API Providers.

To create an API Provider

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **API provider**.
3. Click **Create** in the Manage API Providers page.
4. Type a name for the API Provider in the **Name** field.
5. Type a description for the API Provider in the **Description** field.
6. To associate APIs with the API Provider, click Add and select the required APIs from the list.

   Only the APIs that are associated with the Default Provider are listed here.
7. To send an email notification select the email alias of the recipients group from the drop down. This is a user group in UMC.
8. To configure an HTTP callback notification, provide the required URL in the format `http://<host>:<port>/callbackurl`. This URL should accept HTTP POST call.

   a. If the HTTP callback is password protected, provide the **Username** and **Password**.
9. Click **Create**.

   The API Provider is created and listed in the Manage API Providers page.

Deleting an API Provider

You must have the privileges of an API Portal Administrator or an API Provider to delete API Providers.

You cannot delete providers created from CentraSite, API Gateway, and default API Provider in the system.

To delete an API Provider

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **API provider**.
3. Click for the selected API Provider.
4. Click **Yes** in the confirmation dialog.

5. In the confirmation dialog box, select the option to delete the APIs along with the API provider.

   If you do not select the option only the API provider gets deleted and the APIs associated with this Provider get associated with the Default Provider.

6. Click **Yes**.

Modifying Details of an API Provider

You must have the privileges of an API Portal Administrator or an API Provider to modify the details of API Providers.

You can not modify the details of providers created from CentraSite, API Gateway. For the Default Provider you can update only the email notification group.

To modify details of an API Provider

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **API provider**.

3. Click the for the API provider whose details are to be modified.

4. Modify the required details like name, description, associated APIs, and the notifications configured.

5. Click **Apply**.

   The API Provider is updated with the modified information and listed in the Manage API Providers page.

6 Managing API Assets

Planning for API Management

API Portal functions as the key component of an effective API management solution, in combination with the products, CentraSite and API Gateway. CentraSite and API Gateway use API Portal to securely publish APIs to external developers and partners and provides design-time governance capabilities to the APIs, whereas API Portal allows developers to self-register, learn about these APIs, and use the APIs in their applications.

To prepare to manage the APIs that you plan to make available in API Portal, consider the following questions:

- How many API Portal instances you might need?
- Which organizations might use API Portal?
- Which users in the organization might use API Portal to consume the published APIs?
- Which taxonomies and categories are required to organize the APIs?

About API Portal Assets

For each API Portal, there is an API Portal object registered with the API Provider. An API Portal is associated with an organization. Multiple API Portal instances can share the same organization.

An API can be published to multiple API Portal instances. API Portal is capable of managing APIs published from CentraSite and API Gateway.

When an API is unpublished (removed) from API Portal, its metadata is deleted from API Portal, and the API is no longer available for access.

API Portal Profile in CentraSite

The Service asset type contains a profile named API Portal Information that includes attributes that are of use when CentraSite is integrated with API Portal.

The API Portal Information profile includes the following attributes:

- **API maturity status.** Defines the maturity of your API based on a customizable set of terms, allowing you to indicate the maturity status for the API. For example, Beta, Experimental, Test, or Production.
- **API grouping.** Groups the APIs by a definable business terminology to indicate the API usage. For example, CRM; Financial, Banking, and Insurance; Sales and Ordering.
- **API subscription terms.** Specifies the category of the key assigned to the client to access the API based on subscription plans. For example, Donationware, Flat Fee, Pay per use.
- **API icon.** Specifies the icon shown in API Portal to represent the API.
- **Supported access token types.** Specifies the client authentication: API Key or OAuth2.

- **API Details on API Portal(s).** List of URLs generated for API's details on the actual API Portal to which the API is published to.

CentraSite provides a number of standard taxonomy categories that you can use to indicate the maturity status, grouping, and subscription terms for an API or you can create your own custom categories. For information about taxonomies and adding a category, see *CentraSite User's Guide*.

API Portal Information profile is enabled for the Service asset types (Service, REST Service, and XML Service) and its variants (Virtual Service, Virtual REST Service, and Virtual XML Service) only when API Portal Gateway is created.

Publishing and Unpublishing APIs to and from API Portal

To publish APIs from API Gateway or CentraSite to API Portal, you must have the privileges of an API Administrator, or API Provider.

For information about roles and permissions required to publish and unpublish APIs to and from API Portal, see *CentraSite User's Guide* and *webMethods API Gateway User's Guide*.

API Portal uses the following built-in actions for design/change-time policies to facilitate publishing and unpublishing APIs to and from API Portal:

- **Publish to API Portal.** This action bundles API metadata in the configured database, and then publishes those bundles to API Portal. You can assign APIs to private communities as required.
- **Unpublish from API Portal.** This action removes the specified API metadata bundle from the API Portal.

Handling Events

Starting from 9.12 version, API Portal logs the events related to access token like key request, key renew, and key revoke in an events table. The events table also handles events related to user update and user delete.

The events mechanism has two approaches namely, Push and Pull. In the Push mechanism, the event, once saved in events table, is published to the callback URL of the key provider, whereas, in the Pull mechanism the key provider queries the events from the events table.

APIs published from CentraSite

When an API developer signs up and requests an API access token, API Portal logs the request in the local database events table with the status as NEW and sends the API developer's sign-up or key request to CentraSite (when CentraSite is in the online mode). If CentraSite is in the offline mode, a scheduler in CentraSite polls the events database and gets the list of requests or events that are in NEW state, once it is online. Once the queried event reaches CentraSite, the status changes to INPROGRESS in the events table of API Portal.

CentraSite then generates a key, sends the key through an email to the requesting API developer and deploys the key to Mediator. Once the successful request reaches API Portal, the status of the event in the events table changes from INPROGRESS to COMPLETED. If the request is rejected

in CentraSite, then the status of the event in the events table changes from INPROGRESS to REJECTED.

On receiving the email, the API developer includes the key in the application so that, at run time, when the application communicates with the virtual service at the Mediator target endpoint, Mediator calls the native service. Mediator acts as a key enforcer validating the key contained in the application's header. In addition to key validation, Mediator also collects metrics about the application and sends that data to CentraSite for analysis and displaying the information on dashboards.

Note:
For API Portal 9.12 configured with CentraSite 9.10, the key request handling for API access tokens for the APIs published from CentraSite 9.10 to API Portal 9.12, works the same way as it did when the API was published to API Portal 9.10, except for the introduction of events table.

Querying the requests that have the status as NEW

Method: GET

EndPointURL: http://<*host*>:<*port*>/ abs/apirepository/v1/events?eventStatus=NEW

Updating the status of an event after processing

Method: POST

EndPointURL: http://<*host*>:<*port*>/ abs/apirepository/v1/events

Request payload

```
[
 {
        "eventId":"<eventId>",
        "eventStatus":"ACCEPTED/REJECTED"
 }
]
```

If the response event status is ACCEPTED then the status for the request in events table is updated to INPROGRESS, else it is moved to the status REJECTED. Events in INPROGRESS state can be queried by providers and the request processed. Once the event id is processed it moves to the COMPLETED state. The events in REJECTED and COMPLETED states are cleared based on the value provided for **Events expiry days** variable in the **Administration > Configuration Settings** page.

API Portal Extension Points

Starting from 9.12 version, API Portal works as a standalone product where any third-party provider can be integrated as a key management provider. The third-party key provider should register as a provider in API Portal. You can publish a protected API and link it with the provider.

When a user sends the requests such as new key request, renew request or revoke request, the requests are stored in a local database in an events table with the status as NEW in API Portal. If the third-party provider is configured with the Push mechanism, the request is immediately sent to the provider and the status is changed to INPROGRESS in the events table. If the third-party

provider is configured with Pull mechanism, then it starts querying events that have the status as NEW from the events table in API Portal. Once the event receipt is acknowledged by third-party provider, the status is changed to INPROGRESS in events table. Once the third-party provider processes the request and the success response is sent back to API Portal, the status changes to COMPLETED in the events table. In case the request is rejected, then the status is updated as REJECTED.

Managing Third-party Key Management Providers

As part of managing key management providers you can register Providers, link and unlink APIs with Providers, and create protected APIs.

An external Provider can be registered in the following ways:

- With call back URL
- With SMTP details
- Without cal back URL and SMTP configuration

Registering a Provider with call back URL

HTTP Method: POST

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/providers

Payload

```
{
  {
  "apiportaluuid": "e500b8bb-26e5-4963-98c6-246789a2fc96",
  "providerurl": "http://sag.cs.org/external",
  "version": "9.12.0.3”
  "notificationPlugins": [
    {
      "id": "HTTP",
      "pluginParameters": {
        "com.aris.umc.apiportal.external.event.notify.endpoint":
        "http://sag.cs.org:53307/Centrasite/apimgmt/accesstokens/v1",
        "com.aris.umc.apiportal.external.event.notify.method": "POST",
        "com.aris.umc.apiportal.external.event.notify.contentType": "application
         /json"
      }
    }
  ]
}
```

Usage: When consumers request, renew or revoke access tokens from API Portal, the request is sent to the POST endpoint that publishes the token to API Portal.

Registering a Provider with SMTP details

HTTP Method: POST

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/providers

Payload

```
{
  "name": "Test Provider",
  "short Description": "External Gateway",
  "longDescription": "An external non webmethods gateway which will poll for new
API subscriptions",
  "providerurl”: "",
  "notificationPlugins": [
    {
      "id": "SMTP",
      "pluginParameters": {
        "com.aris.umc.apiportal.external.event.notify.subject"
         :"New ${event.type}",
        "com.aris.umc.apiportal.external.event.notify.message":"Hello,/n/nThere
         is a new ${event.type} event received from webMethods API Portal
         (${event.portalURL}) /n /n Best Regards, /n /n API Portal Team/n/n***
          This notification was sent automatically.
          Do not reply to this email.***"
          }
    }
  ]
}
```

Usage: When consumers request, renew or revoke access tokens from API Portal, the request details are mailed to the provider's mail id. The provider publishes the token to API Portal with the proper endpoint.

Registering a Provider without call back URL and SMTP configuration

HTTP Method: POST

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/providers

Payload

```
{
  "name": "Test Provider",
  "short Description": "External Gateway",
  "longDescription": "An external non webmethods gateway which will poll for
new API subscriptions",
  "providerurl”: ""
}
```

Usage: When consumers request, renew or revoke access tokens from API Portal, the event is stored in events database (in NEW state). The provider has to query the events and should update the status of the event.

Linking an API with a Provider

HTTP Method: PUT

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/providers/<providerId>/apis

Payload

```
[
  "api1_id",
  "api2_id"
]
```

Example:

```
[
  "496ef631-23cf-11e6-4593-d4bed967adb3",
   "ac90c761-1e5d-11e6-3117-d4bed967adb3"
]
```

Creating a protected API

HTTP Method: PATCH

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/apis/{api_id}

Payload

```
[
   {
      "op":"replace",
      "path":"isProtected",
      "value":"true"
   }
]
```

Unlinking an API with a Provider

HTTP Method: DELETE

EndPoint URL: http://<*host*>:<*port*>/abs/apirepository/providers/<providerId>/apis

Payload

```
[
  "api1_id",
  "api2_id"
]
```

Example:

```
[
  "496ef631-23cf-11e6-4593-d4bed967adb3",
   "ac90c761-1e5d-11e6-3117-d4bed967adb3"
]
```

When an API is linked with a provider (linked by means of call back URL method), then the request is processed as the provider has implemented the API.

When an API is linked with a provider (linked by means of SMTP notification method), then the request details are mailed to the provider's e-mail id. The provider then processes the request with the event-id received in the mail.

When an API is linked with a provider, who is configured without the call back URL or SMTP notification, then the provider has to query the event and process it.

When an API is not linked with any Provider, then the request details are mailed to the users of API Consumption Approver group in UMC. One of the user from that group processes the request with event-id received in the mail.

Managing Access Tokens

This section explains how a provider registered with SMTP details can provide access tokens for the consumers.

Accessing the list of all events in the INPROGRESS status

Method: GET

EndPoint: http://<*host*>:<*port*>/abs/apirepository/v1/events?eventStatus=INPROGRESS

Publishing an access token

A provider can publish the access token for the respective access token request based on the eventId value.

Method: POST

EndPointURL: http://<*host*>:<*port*>/ abs/apirepository/v1/accesstokens?eventId=<eventId>

Sample request payload for API key

```
{
    "name": "<application name>",
    "description": "<sample desc>",
    "appId": "65177516-3fd8-47ea-89b5-6d94eb2ce5c0",
    "userId": "4ee9c60c-c32e-3615-83c4-5dc6822c45de",
    "type": "Application",
    "accessTokenDetails": [{
        "type": "APIKey",
        "properties": [{
            "key": "apiKeyString",
            "value": "97cab587-2a3c-4308-8268-b636ea1479b6"
        }]
    }]
}
```

Sample request payload for OAuth2

```
{
    "name": "<application name>",
    "description": "<sample desc>",
    "appId": "51051137-2de6-44ad-98ef-f3761eb5ca59",
    "userId": "4ee9c60c-c32e-3615-83c4-5dc6822c45de",
    "type": "Application",
    "accessTokenDetails": [{
            "type": "OAuth2",
            "properties": [{
```

```
                    "key": "clientId",
                    "value": "18c2190c-dc8e-449f-aa09-9cdda75bf4d0"
                },
                {
                    "key": "clientSecret",
                    "value": "09d37e9d-d16c-49b6-a436-64e806f188cc"
                },
                {
                    "key": "refreshCount",
                    "value": "0"
                },
                {
                    "key": "expiryDate",
                    "value": "3600"
                },
                {
                    "key": "redirect_uris",
                    "value": "[\"<redirect_uri endpoint>\"]"
                },
                {
                    "key": "authorization_uris",
                    "value": "[\"<authorization_uri endpoint>\"]"
                },
                {
                    "key": "accesstoken_uris",
                    "value": "[\"<accesstoken_uri endpoint>\"]"
                },
                {
                    "key": "refreshtoken_uris",
                    "value": "[\"<refreshtoken_uri endpoint>\"]"
                }
            ]
        }
    ]
}
```

Sample response payload for the API key once the request is processed successfully

```
{
   "status":"SUCCESS",
   "message":"Access token created successfully!",
   "accesstokenuuid":"<GUID>"
}
```

Renewing an access token

The renew access token request is based on the `accesstokenuuid` and `eventId value`.

Method: PUT

EndPointURL: http://<*host*>:<*port*>/
abs/apirepository/v1/accesstokens/<accesstokenuuid>?eventId=<eventId>

Request payload

```
{
   "type":"APIKey",
   "accessTokenDetails":[
```

```
      {
         "key":"apiKeyString",
         "value":"< apiKeyString >"
      },
      {
         "key":"expiryDate",
         "value":"< expiryDate >"
      }
   ]
}
```

Response payload

```
{
   "status":"SUCCESS",
   "message":"Access token renewed successfully!"
}
```

Revoking an access token

The revoke access token request is based on the `accesstokenuuid` and `eventId value`.

Method: DELETE

EndPointURL: http://*<host>*:*<port>*/
abs/apirepository/v1/accesstokens/<accesstokenuuid>?eventId=<eventId>

Response: 204 (No content)

7 Managing APIs

Manage APIs

You can manage APIs from the Manage APIs page. The page lists all the APIs and their description. You can edit an API, delete an API, and import an API from this view.

Field	Description
Name	The name of the API.
Description	The description about the API.
Version	The version number of the API.
Operations	Operations performed on the API. Available options are: ■ Delete ■ Edit

Importing an API Directly through the API Portal User Interface

API Portal offers direct API importing facility for standalone scenarios when you do not have CentraSite or Mediator installed.

You can edit or delete the directly imported APIs as required. The supported API formats are **RAML**, **Swagger**, and **WSDL**.

You should have the API Provider or API Portal Administrator privileges to directly import an API. You can import the API in one of the following ways:

- "Importing an API by Uploading an API" on page 134
- "Importing an API by Providing an API URL" on page 136
- "Importing an API by Copying and Pasting API Content" on page 137

Importing an API by Uploading an API

API Portal offers direct API importing facility for standalone scenarios when you do not have CentraSite or Mediator installed.

Prerequisite

You must have the **API Provider** or **API Administrator** role.

To import an API by uploading an API

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage APIs**. The **Manage APIs** page that lists all the APIs is displayed.
3. Click **Import API**.
4. Click **Upload API**.
5. Click **Browse**.
6. Select the required file and click **Open**.

   The Swagger parser is a self-contained file with no external references and can be uploaded as is.

   If the RAML file, that is to be imported, contains external references, the entire set of files must be uploaded as a zip file with a structure as referenced by the RAML file.

   For WSDL, if it is a single .wsdl file, it can be uploaded as it is. If the wsdl file contains reference schema, the entire set of files must be uploaded as a zip file with a structure as referenced by the WSDL file. If there are multiple .wsdl files in the zip file, then you have to provide the root file name.
7. Select the type.

   The available types are **OpenAPI**, **RAML**, **WSDL**, or user-defined. By default, the options available are **RAML**, **OpenAPI**, and **WSDL**.
8. Type the API name you want the API to be displayed as.

   If you provide an API name, this overwrites the API name mentioned in the uploaded file and imported API is displayed with the name provided.

   If you do not provide an API name, the API name mentioned in the uploaded file is picked up and the API is displayed with that name.

   If you do not provide an API name and the uploaded file does not have an API name mentioned then the API is displayed as **Untitled**.
9. Select the API Provider.

   The imported API is associated with the API Provider selected. If a provider is not selected, by default the API is associated with the Default Provider.
10. Click **Import API**.
11. Click **Close**.

The imported API is now listed in the list of APIs.

You can use the **Delete** option to delete an API and the **Edit** option to update the API. The **Edit** option is available only for the directly imported APIs.

Importing an API by Providing an API URL

API Portal offers direct API importing facility for standalone scenarios when you do not have CentraSite or Mediator installed.

Prerequisite

You must have the **API Provider** or **API Administrator** role.

To import an API by providing an API URL

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage APIs**. The **Manage APIs** page that lists all the APIs is displayed.
3. Click **Import API**.
4. Click **API URL**.
5. Type the required URL.
6. Select the type.

   The available types are **OpenAPI**, **RAML**, **WSDL**, or user-defined. By default the options available are **RAML**, **OpenAPI**, and **WSDL**.
7. Type the API name you want the API to be displayed as.

   If you provide an API name, this overwrites the API name mentioned in the uploaded file and imported API is displayed with the name provided.

   If you do not provide an API name, the API name mentioned in the uploaded file is picked up and the API is displayed with that name.

   If you do not provide an API name and the uploaded file does not have an API name mentioned then the API is displayed as **Untitled**.
8. Select the API Provider.

   The imported API is associated with the API Provider selected. If a provider is not selected, by default the API is associated with the Default Provider.
9. Click **Import API**.
10. Click **Close**.

The imported API is now listed in the list of APIs.

You can use the **Delete** option to delete an API and the **Edit** option to update the API. The **Edit** option is available only for the directly imported APIs.

Importing an API by Copying and Pasting API Content

API Portal offers direct API importing facility for standalone scenarios when you do not have CentraSite or Mediator installed.

Prerequisite

You must have the **API Provider** or **API Administrator** role.

To import an API by copying and pasting API content

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage APIs**. The **Manage APIs** window that lists all the APIs is displayed.
3. Click **Import API**.
4. Click **API editor**.
5. Paste the required parser content in the text box.

   The content should not have any references to an external file.
6. Select the type.

   The available types are **OpenAPI**, **RAML**, **WSDL**, or user-defined. By default, the options available are **RAML**, **OpenAPI**, and **WSDL**.
7. Type the API name you want the API to be displayed as.

   If you provide an API name, this overwrites the API name mentioned in the uploaded file and imported API is displayed with the name provided.

   If you do not provide an API name, the API name mentioned in the uploaded file is picked up and the API is displayed with that name.

   If you do not provide an API name and the uploaded file does not have an API name mentioned then the API is displayed as untitled.
8. Select an API Provider.

   The imported API is associated with the API Provider selected. If a provider is not selected, by default the API is associated with the Default Provider.

9. Click **Import API**.

10. Click **Close**.

The imported API is now listed in the list of APIs.

You can use the **Delete** option to delete an API and the **Edit** option to update the API. The **Edit** option is available only for the directly imported APIs.

Deleting an API

You must have the **API Provider** or **API Administrator** role.

To delete an API

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **Manage APIs**.

   The Manage APIs page that lists all the APIs is displayed.

3. Click for the API you want to delete.

You deleted an API.

Updating an API

The **Update** option is available only for the directly imported APIs.

Prerequisite

You must have the **API Provider** or **API Administrator** role.

To update an API

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **Manage APIs**. The **Manage APIs** window that lists all the APIs is displayed.

3. Click for the API you want to update.

4. Update the API by uploading a new file or by providing a new URL or pasting new content.

5. Click **Update**.

   The API with the updated information is now available in the Manage APIs list.

API Versions

API Portal supports versioned APIs that are published from API Gateway and CentraSite. You can also create a new version of an API when you update an API in API Portal.

When you create a new version of an API and publish it to API Portal, the API Gallery page displays the latest version of the API with the indication that it is versioned. The Manage APIs page displays all the API versions. The API details page has a drop down list that displays all the API versions present along with the maturity status of the API, if any. Selecting a version from the list displays the API details of that version.

If you have multiple versions of an API, the access token request is enabled only for the latest version. Access tokens of an earlier version can be used to test newer versions of the API.

Different versions of an API can be added to different communities and can be associated with different packages.

Creating a New API Version

You can create a new version of an API using the **Edit** option. This option is available only for the directly imported APIs.

Prerequisite

You must have the **API Provider** or **API Administrator** role.

To create a new API version

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage APIs**. The **Manage APIs** window that lists all the APIs is displayed.
3. Click for the API you want to edit.
4. Update the API by uploading a new file or by providing a new URL or pasting new content.
5. Select **Create as new version**.
6. Click **Update**.

   The new version of the API created is now available in the API Gallery and the Manage APIs list.

Editing APIs

You can edit an API from the API Details page. There are two types of edits you can apply to an API:

- Basic: Involves modifying API name, API description, resource description, method description, method parameter description, and additional attributes.
- Advanced: Involves modifying the icon, supporting documents, and categories of an API.

Modifying the Basic Attributes of an API

You can modify the basic attributes of an API, in the API Details page, in the view mode as well as by clicking the **Edit** button.

To modify the basic attributes of an API

1. Log in to API Portal.
2. Click **API Gallery**.
3. Click **View details** for an API.
4. In the API Details page, you can do one of the following:
   - Hover over the basic attributes and click the edit icon that appears.
   - Click **Edit**. All the editable fields display the edit icon ; click for the required field.

     On clicking **Edit**, the **Advanced edit** option appears under the What's Next? section, which can be used to edit the advanced attributes of an API.
5. Modify the information as required.
6. Click to save the changes.

   You can click to discard the changes.

Modifying the Advanced Attributes of an API

You can modify the advanced attributes of an API using the Advanced edit option.

To modify advanced attributes using Advanced edit

1. Log in to API Portal.
2. Click **API Gallery**.
3. Click **View details** for an API.

4. In the API Details page, click **Edit**.

   The **Advanced edit** option appears in the What's Next? section.

5. Click **Advanced edit**.

6. In the Advanced API editing dialog box you can modify any or all the following:

   - Click  (API icon). Click **Browse...**, select the required icon, click **Open**, and click **Save**.

     Note:
     Supported filed types are JPG, JPEG, GIF, SVG, and PNG and maximum image size is 1MB.

   - Click  (API documents). Click **Add**, select the required documents, and click **Open**.

     Note:
     Supported files types are PDF, DOC, ZIP, JPG, JPEG, DOCX, XLS, XLSX, PNG, PPT, and PPTX and maximum file size is 2 MB. The supported file type and maximum size parameters are configurable in the Configuration Settings page.

     If you want to delete the existing document you can click the delete icon for the corresponding API document listed.

   - Click  API category. Provide the **Tags**, **Business term**, **Maturity state**, and **API group** as required and click **Save**.

Markdown Support

Markdown is a lightweight language that is used to render HTML content using plain text formatting syntax.

Markdown support is available for the following attributes in API details page:

- API short description
- API long description
- Resource description
- Method short description
- Method long description
- Values in Other Attributes section

Note:
Addition of authentic and valid markdown content is recommended for security reasons.

8 Using and Testing APIs

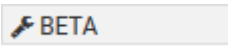
API Gallery

The API Gallery lists all the APIs that are present. If an API has multiple versions only the latest version is displayed. You can view the details of individual API or group the APIs as required.

The APIs listed in the API gallery are one of the following type:

- REST API
- SOAP API
- OData API (API Portal supports Version 2 and Version 4 of OData API.)
- Hybrid API (This is a REST API with the SOAP metadata displayed.)

In the API gallery view you can perform the following operations listed in the table.

Action	Description
	Group APIs as required. You can group them with the following options available: ■ **API group** ■ **Business term** ■ **Communities** ■ **Maturity status** The APIs that do not belong to the group specified are listed as Ungrouped. **Note:** If a user leaves a community or is removed from a community, the API Gallery page takes a couple of minutes to synchronize the changes.
	Displays the details of the selected API. The details such as API description, API Resources, API documents, Access API, API Version, and Latest posts are displayed.
	Indicates that the API is versioned and different versions of the API are available.
	Indicates the business term of the API.
	Indicates the maturity status of the API.

Finding APIs in the API Gallery

You can search for the required APIs in the API Gallery.

To find APIs in the API Gallery

- In the navigation bar, click **API Gallery**.

All APIs are displayed. APIs are structured in categories.

Alternatively, on the **Home** page, you can type the beginning of the API name in the search box. A list of relevant APIs is displayed.

APIs in development or deprecated APIs will be displayed only to API providers or administrators. Public APIs can be viewed by guest users without login.

Note:
APIs that are present in two different stages are displayed as one tile in the API Gallery. However, when you click an API that is present in more than one stage, you can see the list of stages that the API is present in.

Viewing API Details

You can view the details of an API in the API details page.

To view API details

1. Log in to API Portal.
2. Switch to the **API Gallery** page.
3. Click **View details** for an API.

The API Details page opens. The API Details page shows different entries depending on the API type (REST, SOAP, and OData API).

API Details View

The API details view displays the details of the selected API such as API description, API Resources, API documents, Access API and Latest posts. You can also look at the discussion on the support forum regarding this API, choose to follow this API, export the API, rate the API or view the list of followers of the selected API.

If the API is a Hybrid API, by default, it is displayed as a REST API and the REST resources are displayed in the API Details view. You can view the SOAP data by clicking the SOAP option.

Field	Description
API name	Displays the API name and the tags associated with the API.
About *API name*	Displays the name and description of the API. The API name and description of the API are editable. Hovering over the API name or the API description displays the edit icon, which you can click and edit them as required.
API resources (For a REST API)	Displays a list of resources available in the API sorted by resource/pathname. For the REST APIs, the list of resources are displayed in sorted order of the path names. Click each resource to view the corresponding HTTP methods, along with a summary. For each of these methods, details such as tags, parameters and status codes, schema definition, callback URLs, sample request, and sample response with links are displayed. If a method is protected by APIKey, OAuth2 or JWT, the same is indicated by a lock symbol.
API methods (For a SOAP API)	Displays a list of methods available in the API. For the SOAP APIs, methods are displayed along with their type of binding (SOAP 11 , SOAP 12, and other HTTP methods). Click each method to view details such as input, output, fault messages, and tags. If a method is protected by APIKey, OAuth2 or JWT, the same is indicated by a lock symbol.
API components For REST APIS created with Open API specifications	Displays a list of schema available in the API. Click each schema to view more details You can click on each schema to view the schema details, such as name, description, type, example, whether it is required and array information in the Table tab and the schema format in the JSON tab
Other attributes	Displays a list of any other attributes associated with the API. For example, Contact name, License URL, License name, Contact email, and so on.
Entity sets (For an OData API)	An Entity Set element represents a single entity or a collection of entities of a specific entity type in the data model. Type of the entity and the container name are displayed.
Singletons	Singletons are single entities which are accessed as children of the entity container.

Field	Description
(For an OData API)	Type of the singleton and the container name are displayed.
Entity types (For an OData API)	Entity Types (for example, Person, Airline and so on) are structured records consisting of named and typed properties and key properties whose values (for example, UserName, AirlineCode and so on) uniquely identify one instance from another. Here the list of available entity types and the parameters are displayed.
Complex types (For an OData API)	Complex Types are structured types (for example, City, Location, Airport Location and so on) consisting of a list of properties (for example, CountryRegion, Name, Address, and so on) but with no key, and thus can only exist as a property of a containing entity or as a temporary value. List of available complex types and the parameters are displayed.
Functions (For an OData API)	The Function element represents a parameter to the function. Function renders the parameters that are accepted by the function and return type of the function.
Actions (For an OData API)	The Action element denotes if the action is bound to a specific entity type in an entity model. Displays the parameter section that is rendered in functions.
API documents	Displays the list of all documents for the API.
Access API	Displays the list of endpoints for the API.
API scopes (For a SOAP and REST API)	A scope represents a logical grouping of REST resources, methods, or both, and SOAP operations in an API. You can then enforce a specific set of policies on each individual scope in the API. Displays the scopes associated with the API and the corresponding details such as name of the scope, description of the scope and the methods included in the scope. If the scope has policies associated, the policies are displayed under a subsection Policies.
Latest posts	Opens a text editor where you can type in your comment, tag the post, attach a link or a file and post it.
Try API	Click this option to test the API. It opens the Try API page. You can provide the various parameters required for the API and test it.

Field	Description
	For a REST API, the REST resources that can be tested are displayed in the left panel. For a SOAP API, the API methods that can be tested are displayed in the left panel. For a Hybrid API, you can click the REST toggle option, in the left panel, to display the REST resources to be tested and click the SOAP toggle option to display the SOAP methods that can be tested.
Version	Displays the version of the API along with the maturity status, if any, and lists all the versions of the API from the latest to the least in the drop-down list. Selecting a version from the list displays the API details of that version.
Support forum	Clicking this option navigates you to the Collaboration > My feed view where you can view the recent posts or activity. You can also post it to your feeds.
Get access token	Clicking this option you can request an access token to access and use the API. This option is available only for the latest API version.
Follow this API	Click to follow this API. If you are already following this API you will see the option Unfollow this API, in which case you can click it to unfollow the API.
Export API as (For a REST API)	Click to export the API in the JSON or YAML format. The schemas of the API must comply with at least Draft 4 version of JSON Schema.
Download Client SDK (For a REST API)	Click to download the Client SDK for the API in the specified language. **Supported languages** - Akka-Scala, Android, ASP.NET5, Async-Scala, Clojure, C++Rest, C#, C#.NET2, Cwiki, Dart, Dynamic-Html, Flash, Go, Go-Server, Groovy, Haskell, HTML, HTML2, Inflector, Java, Javascript, Javascript-Closure-Angular, JAXRS, JAXRS-CXF, JAXRS-Rest, JAXRS-Spec, JMeter, Lumen, Nancyfx, Nodejs-Server, Objective C, Perl, Php, Python, Python-Flask, Qt5 C++, Rails5, Ruby, Scala, Scalatra, Silex-Php, Sinatra, Slim, Spring, Swagger, Swagger-Yaml, Swift, Tizen, Typescript-Angular, Typescript-Angular2, Typescript-Fetch, Typescript-Node.

Field	Description
	The schemas of the API must comply with at least Draft 4 version of JSON Schema. This is available only for REST APIs.
Advanced edit	Click to modify the icon representing the API, API documents and API category for the selected API.
Rate this API	Click the number of stars depending on how you want to rate this API. Rating is specific to an API version.
List of followers	Displays the number of followers who are following this API. Followers of a previous API version automatically follow any subsequent higher versions. **Note:** The list of followers is not persisted in the following cases : ■ A SOAP API republished as REST API or a Hybrid API. ■ A REST API or a Hybrid API republished as a REST API.
Associated packages	Displays the associated packages with the API.

Editing APIs

You can edit an API from the API Details page. There are two types of edits you can apply to an API:

- Basic: Involves modifying API name, API description, resource description, method description, method parameter description, and additional attributes.
- Advanced: Involves modifying the icon, supporting documents, and categories of an API.

Modifying the Basic Attributes of an API

You can modify the basic attributes of an API, in the API Details page, in the view mode as well as by clicking the **Edit** button.

To modify the basic attributes of an API

1. Log in to API Portal.
2. Click **API Gallery**.
3. Click **View details** for an API.

4. In the API Details page, you can do one of the following:

   - Hover over the basic attributes and click the edit icon that appears.
   - Click **Edit**. All the editable fields display the edit icon ; click for the required field.

     On clicking **Edit**, the **Advanced edit** option appears under the What's Next? section, which can be used to edit the advanced attributes of an API.

5. Modify the information as required.

6. Click to save the changes.

   You can click to discard the changes.

Modifying the Advanced Attributes of an API

You can modify the advanced attributes of an API using the Advanced edit option.

To modify advanced attributes using Advanced edit

1. Log in to API Portal.
2. Click **API Gallery**.
3. Click **View details** for an API.
4. In the API Details page, click **Edit**.

   The **Advanced edit** option appears in the What's Next? section.
5. Click **Advanced edit**.
6. In the Advanced API editing dialog box you can modify any or all the following:

   - Click (API icon). Click **Browse...**, select the required icon, click **Open**, and click **Save**.

     Note:
     Supported filed types are JPG, JPEG, GIF, SVG, and PNG and maximum image size is 1MB.

   - Click (API documents). Click **Add**, select the required documents, and click **Open**.

     Note:

Supported files types are PDF, DOC, ZIP, JPG, JPEG, DOCX, XLS, XLSX, PNG, PPT, and PPTX and maximum file size is 2 MB. The supported file type and maximum size parameters are configurable in the Configuration Settings page.

If you want to delete the existing document you can click the delete icon for the corresponding API document listed.

- Click  API category. Provide the **Tags**, **Business term**, **Maturity state**, and **API group** as required and click **Save**.

Markdown Support

Markdown is a lightweight language that is used to render HTML content using plain text formatting syntax.

Markdown support is available for the following attributes in API details page:

- API short description
- API long description
- Resource description
- Method short description
- Method long description
- Values in Other Attributes section

Note:
Addition of authentic and valid markdown content is recommended for security reasons.

Testing a REST API

To test a REST API

1. Switch to the **API Gallery** page.

   Alternatively go to the **Home** page and type the beginning of the API name into the search box.

2. Click **View details** for an API.

3. Click **Try API** in the **API details** page.

   The **Try API** page is displayed.

4. Select the required stage of the API that you want to test, from the **Environment** field.

The applications associated with the selected stage are displayed in the **Applications** field.

Note:
This step is applicable only if the API is available in more than one stage.

5. Select an Application to test.

   The **Application** drop-down list displays the applications for a resource or API. The applications are grouped under their respective authentication mechanisms. For example, the applications for OAuth are grouped under the head, OAuth and so on. The resources that can be tested are displayed in the left panel.

   The **Sandbox** drop-down list displays the different instances or stages of API Gateway from which the API is published to API Portal. For example, if an API is published from Dev and Production stages, then both the stages are displayed in the Sandbox drop-down list.

6. Select a resource.

   The methods associated with the resource that can be tested are displayed. If a method is protected by basic authentication, APIKey, OAuth2, or JWT, the same is indicated by a lock symbol.

   Note:
   The HTTP PATCH support is available in API Portal 9.12 onwards.

7. Click a method.

8. Provide the following required values in the **Parameters** tab:

   - Type a value for the path parameters.
   - Type a query parameter and its corresponding value.

     Note:

     You can add multiple entries by clicking + .

9. Select the required Authorization Type from the **Authorization** tab and provide the corresponding values:

   - **No Auth**: Specifies that no authentication is required.
   - **Basic authentication**: Specifies basic authentication of user credentials is required. Type the username and password for authentication and click **Update**.
   - **Oauth 2.0**: Specifies OAuth authentication is required. Provide the required information to fetch a OAuth token to use for authentication. For details on how to generate a OAuth token and use it see, "OAuth 2.0 Support" on page 156.

- **JWT**: Specifies a JSON Web token is required for authentication. Type a token or select one of the available tokens in the list and click **Update**. For details on how to generate a JSON web token and use it see, "Testing a JWT protected API" on page 158.

10. Type in the required Header name(s) and corresponding value(s) in the **Headers** tab. You can add multiple entries by clicking  .

11. Click **Send**.

The request and the response of the API are displayed.

To clear the values entered and response rendered, click **Clear**.

Testing a SOAP API

To test a specific SOAP API proceed as follows.

To test a SOAP API

1. Log in to API Portal.
2. Switch to the **API Gallery** page.

   Alternatively go to the **Home** page and type the beginning of the API name into the search box.
3. Click **View details** for an API.
4. Click **Try API** in the API details page.

   The **Try API** page is displayed.
5. Select the required stage of the API that you want to test, from the Environment field.

   The applications associated with the selected stage are displayed in the Applications field.

   Note:
   This step is applicable only if the API is available in more than one stage.
6. Select an Application to test.

   The methods that can be tested are displayed in the left panel. If a method is protected by basic authentication, APIKey, OAuth2 or JWT, the same is indicated by a lock symbol.
7. Select a method.

   The methods associated with the resource with the corresponding SOAP versions that can be tested are displayed.

8. Select the required Authorization Type from the **Authorization** tab and provide the corresponding values:

   - **No Auth**: Specifies that no authentication is required.
   - **Basic authentication**: Specifies basic authentication of user credentials is required. Type the username and password for authentication and click **Update**.
   - **Oauth 2.0**: Specifies OAuth authentication is required. Provide the required information to fetch a OAuth token to use for authentication. For details on how to generate a OAuth token and use it see, "OAuth 2.0 Support" on page 156.
   - **JWT**: Specifies a JSON Web token is required for authentication. Type a token or select one of the available tokens in the list and click **Update**. For details on how to generate a JSON web token and use it see, "Testing a JWT protected API" on page 158.

9. Type in the required Header name(s) and corresponding value(s) in the **Headers** tab. You can add multiple entries by clicking  .

10. Type the web service user name and password in the **Web service security** tab.

11. Click **Send**.

The request and the response of the API are displayed.

To clear the values entered and response rendered, click **Clear**.

Testing an OData API

To test a specific OData API proceed as follows.

To test an OData API

1. Switch to the **API Gallery** page.

   Alternatively go to the **Home** page and type the beginning of the API name into the search box.

2. Click **View details** for an API.

3. Click **Try API** in the API details page.

   The **Try API** page is displayed.

4. Select the required stage of the API that you want to test, from the Environment field.

   The applications associated with the selected stage are displayed in the Applications field.

Note:
This step is applicable only if the API is available in more than one stage.

5. Select an Application to test.

   The resources that can be tested are displayed.

6. Select a resource.

   The methods associated with the resource that can be tested are displayed.

7. Click a method for the listed **Entity sets** in the left navigation paths and select the method for the entities listed under **Entity sets**.

8. Provide the following required values in the **Parameters** tab:

   - Type a value for the path parameters.
   - Type a query parameter and its corresponding value.

   Note:

   You can add multiple entries by clicking + .

9. Select the required Authorization Type from the **Authorization** tab and provide the corresponding values:

   - **No Auth**: Specifies that no authentication is required.
   - **Basic authentication**: Specifies basic authentication of user credentials is required. Type the username and password for authentication and click **Update**.
   - **Oauth 2.0**: Specifies OAuth authentication is required. Provide the required information to fetch a OAuth token to use for authentication. For details on how to generate a OAuth token and use it see, "OAuth 2.0 Support" on page 156.
   - **JWT**: Specifies a JSON Web token is required for authentication. Type a token or select one of the available tokens in the list and click **Update**. For details on how to generate a JSON web token and use it see, "Testing a JWT protected API" on page 158.

10. Type in the required Header name(s) and corresponding value(s) in the **Headers** tab. You can add multiple entries by clicking + .

11. Click **Send**.

The request and the response of the API are displayed.

To clear the values entered and response rendered, click **Clear**.

OAuth 2.0 Support

The OAuth 2.0 Authorization Framework (OAuth) facilitates the sharing of private resources (data or services) with a third-party client application (client). In an OAuth session, private resources are stored on a resource server and the owner of the resources, or resource owner, grants the client application permission to access them. In this case the resource owner is typically an application. When a resource owner grants permission, the OAuth authorization server issues an access token to the client application. When the client application passes the access token to the resource server, the resource server communicates with the authorization server to validate the token and, if valid, provides access to the resources.

Note:
Extensive OAuth2 support is available from API Portal 10.1 and higher versions.

Authorization grant types supported in API Portal

The flow of authorization requests and responses between the resource owner, client application, authorization server, and resource server depends on the authorization grant type defined by the OAuth session. API Portal supports the following authorization grant types:

- Authorization code
- Implicit
- Client credentials

Note that:

- For an OAuth2 protected API published from CentraSite, only Client credentials grant type is supported.
- All 3 grant types are supported only if the OAuth2 protected API is published from API Gateway 10.1 or higher versions.

Authorization Code Grant

The authorization code grant type is used to authenticate and provide access to clients that have credentials on the authorization server. This grant type requires the client to authenticate to the authorization server before obtaining an access token. The user is prompted to log on to the authorization serve and to authorize or deny the client (application) access to their account. Once authorized, the Authorization provides an authorization code. This authorization code is provided to the Access token URL, which in turn provides the required access token to access the resource.

Implicit Grant

The implicit grant flow is similar to the authorization code grant flow with a distinct difference that the Authorization server provides the access token instead of authorization code. The Client ID and Client secret is passed on to the Authorization URL, for validation. The user is prompted to log on to the authorization server and approve the client. On approval, the required access token is provided to access the resource.

Client Credentials

The simplest of all of the OAuth 2.0 grants, this grant is suitable for machine-to-machine authentication where a specific user's permission to access data is not required. The Client ID and Client secret is passed on to the Access token URL, which in turn provides the required access token to access the resource.

Testing an OAuth Protected API

When an OAuth protected API is published from API Gateway to API Portal you require an OAuth2 access token to test the API. The Authorization server can be configured to use either HTTP or HTTPS connection to authorize requests. For information on enabling HTTPS mode for Authorization server in API Gateway see, *API Gateway Administrator's Guide* and *webMethods Integration Server Administrator's Guide*.

To test an OAuth2 protected API

1. Switch to the **API Gallery** page.

   Alternatively, go to the **Home** page and type the beginning of the API name in the search box.

2. Click **View details** for an API.

   The API Details page opens.

3. Click **Get access token** if you are accessing the API for the first time, else proceed to step 6.

4. In the Request API access token dialog box, provide the **Application name** and **Application description**.

   The application is created and listed in the Applications page.

5. Click **Request token**.

6. In the API details page, click **Try API**.

   The application is listed in the Try API page.

7. Type a path parameter key and its value in the respective fields in the **Parameter** tab. You can add multiple entries by clicking  .

8. Provide the following information in the **Authorization** tab:

   - **Authorization Type**: Select OAuth.
   - **Token name**: Type a name for the token.
   - **Grant type**: Select the grant type to be used to authenticate the API. Available values are **Authorization code**, **Implicit**, **Client Credentials**.

The **Authorization URL**, **Access token URL**, **Client Id**, and **Client secret** are pre-populated depending on what grant type is selected.

9. Click **Get token**.

10. In the API Gateway Resource access approval page, click **Approve**.

    This page lists all the APIs that are added as part of the application.

    Note:
    This step is applicable only if you have selected Authorization Code or Implicit in the **Grant type** field.

    A token is generated and is listed under the available token list. An Authorization header is added along with the access token value.

11. Type in the required Header name(s) and corresponding value(s) in the **Headers** tab. You can add multiple entries by clicking  .

12. Select the access token and click **Send**.

    The response is displayed.

    To clear the values entered and response rendered, click **Clear**.

Testing a JWT protected API

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.

When a JWT protected API is published from API Gateway to API Portal you require a JWT access token to test the API.

To test a JWT protected API

1. Switch to the **API Gallery** page.

   Alternatively, go to the **Home** page and type the beginning of the API name in the search box.

2. Click **View details** for an API.

   The API Details page opens.

3. Click **Get access token** if you are accessing the API for the first time, else proceed to step 6.

4. In the Request API access token dialog box, provide the **Application name** and **Application description**.

The application is created and listed in the Applications page.

5. Click **Request token**.
6. In the API details page, click **Try API**.

   The application is listed in the Try API page.
7. Type a path parameter key and its value in the respective fields in the **Parameter** tab. You can add multiple entries by clicking  .
8. Type the API Gateway Administrator credentials in the User name and Password fields in the **Authorization** tab.
9. Click **Get token**.
10. Click **OK**.

    A token is generated and is listed under the available token list. An **Authorization header** is added along with the access token value.
11. Select the access token and click **Send**.

    The response is displayed.

    To clear the values entered and response rendered, click **Clear**.

Following an API

You can follow an API on a social network.

To follow an API

1. Search for APIs or find one in the API Gallery.
2. Click the API to view the API Details page.
3. Click **Follow this API**.

   The list of followers increases by 1. The option now changes to **Unfollow this API**. You can click this to unfollow the API.

You will receive a corresponding notification for any event involving the API.

Sharing an API

You can share an API to let your contacts know about anything interesting you found about the API.

To share an API

1. Search for APIs or find one in the API Gallery.
2. Click the API to view the API Details page.
3. You can share the API in one or more of the following ways:
   - Click  to share the API through facebook. This opens the browser where you have to login into your facebook account and share the API.
   - Click  to share the API through Google. This opens the browser where you have to login into your Google account and share the API.
   - Click  to share the API through email. This opens the compose new mail screen where you type the receivers email address and send the mail to share the API.

The API is now shared.

Downloading Client SDK for an API

You can download the Client SDK for an API in a specified language as a zip file. The zip file contains necessary code package that can be used for testing and communicating with the API. This is available for REST APIs.

To download the client SDK for an API

1. Search for APIs or find one in the API Gallery.
2. Click the API to view the API Details page.
3. Click **Download Client SDK**.
4. Select the required language.

   Supported languages - Akka-Scala, Android, ASP.NET5, Async-Scala, Clojure, C++Rest, C#, C#.NET2, Cwiki, Dart, Dynamic-Html, Flash, Go, Go-Server, Groovy, Haskell, HTML, HTML2, Inflector, Java, Javascript, Javascript-Closure-Angular, JAXRS, JAXRS-CXF, JAXRS-Rest, JAXRS-Spec, JMeter, Lumen, Nancyfx, Nodejs-Server, Objective C, Perl, Php, Python, Python-Flask, Qt5 C++, Rails5, Ruby, Scala, Scalatra, Silex-Php, Sinatra, Slim, Spring, Swagger,

Swagger-Yaml, Swift, Tizen, Typescript-Angular, Typescript-Angular2, Typescript-Fetch, Typescript-Node.

5. Click **Download**.

   The Client SDK is now downloaded.

9 Managing Applications

Applications

An application defines the precise identifiers by which messages from a particular consumer application is recognized at run time. The identifiers can be, for example, user name in HTTP headers, a range of IP addresses, such that API Portal can identify or authenticate the applications that are requesting an API.

The ability of API Portal to relate a message to a specific application enables it to:

- Control access to an API at run time (that is, allow only authorized applications to invoke an API).
- Monitor an API for violations of a Service-Level Agreement (SLA) for a specified application.
- Indicate the application to which a logged transaction event belongs.

Field	Description
Name	The name of the application.
Description	The description of the application.
Environment	The environment from where the application is published. Move the mouse pointer over the icon to view the stage of the application.
Purpose	Specifies the purpose of the application. For example API subscription, Package subscription.
Owner	The name of the user who owns the application.
Share	Option to share the application with a required user or user group (community).
Revoke Access	Option to revoke access to the application from a user or user group.

Viewing List of Applications and Application Details

You can view the list of applications in the Applications page from where you can select an application to view its details. In the Application details page you can see the list of associated APIs and renew or revoke access tokens for those APIs.

To view the application list and application details

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage applications**.

   A list of all registered applications is displayed.

3. Select an application.

   The application details page displays the basic information, access tokens, and APIs registered for that application.

   Note:
   You cannot view the stage from which a package is published, when it is published from API Gateway.

Application Details View

The application details view displays the details of the selected application such as application description, purpose for which the application is created, access tokens, and associated APIs. You can also renew or revoke the access token for an associated API.

Field	Description
Basic information	Provides the basic information of the selected application such as description and purpose for which the application is created. Also, it displays the environment from where the application is published.
Access tokens	Provides the list of access tokens available for the associated APIs for the application. It displays the following details of the access token: ■ **API access key**: The access for using the access token for the associated API. ■ **Expiration date**: The date when the access token expires.
OAuth2 credentials	Provides the OAuth2 credential details available for the application. It displays the following details of the OAuth2 credentials: ■ **Client Id**: Specifies the client identifier issued to the client to identify itself to the authorization server. ■ **Client secret**: Specifies the client secret that is matching to the client identifier. ■ **Token life time (sec)**: Specifies the time for which the access token is available. ■ **Refresh count**: Specifies the number of times the access token can be refreshed.

Field	Description
	■ **Authorization URLs**: Specifies the Authorization URL to which the Client ID and Client secret is passed on for validation. ■ **Access token URLs**: Specifies the Access token URL to which the Client id and Client secret is passed on to and which in turn provides the required access token to access the resource. ■ **Redirect URLs**: Specifies the URIs that the authorization server uses to redirect the resource owner's browser during the grant process. You can add more than one URI at a time using the **Add** button.
JWT properties	Provides the Access token URL details used to get the access token. ■ **Access token URLs**: Specifies the Access token URL that is used, along with the App ID to get an access token from the API provider to access the resource.
Associated APIs	Lists the associated APIs for the selected application. It displays the following information for the associated APIs: ■ **API name**: Specifies the name of the API. ■ **Version**: Specifies the version of the API. ■ **Action**: Displays the following actions that can be performed for the specified API. **Note:** This section is applicable only for application owners. Other users cannot perform these actions on an application. ■ **Renew**: Click to renew the access token for the required API. ■ **Revoke**: Click to revoke an access token for the required API.
Sharing Information	Lists the name and description of users and teams with whom the application is shared with. You can revoke the access of an application from a user or team from this page. **Note:** This section is applicable only for application owners. Other users cannot view this.

Renewing Access Tokens

The renew option is available only for the APIs that are published from CentraSite with an expiry date.

To renew an access token

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Applications**.
3. Click **Renew**, in the Associated APIs section, for the required API.

   The access token is renewed for the specified time.

Revoking Access Tokens

To revoke an access token

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Applications**.
3. Click **Revoke**, in the Associated APIs section, for the required API.

   The access token is revoked and the application is removed from the applications list page, if there is only one API associated to the application.

Application Sharing

You can share an application with a user or a team from the Applications page.

Sharing an Application

To share an application

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Applications**.

   The list of applications available is displayed.
3. Click **Share** next to the application that you want to share.

The Share tab of the Share Settings screen is displayed.

4. Enter the name of the user or team that you want to share the application with.

   The user(s) and team(s) that match the string you enter are displayed.

5. Select the required user or team.

   The selected user or team is displayed. You can repeat this step to add more than one user or team.

6. Click Save.

   The selected application is shared with the user or team. An email is sent to users or teams to notify them the details of the shared application.

The users with whom the application is shared can now access the application shared with them from the Applications page.

Revoking Access from a Shared Application

You can revoke access from a shared application.

To revoke access from an application

1. Click  in the right top corner of the API Portal window to display the menu options.

2. Click **Applications**.

   The list of applications available is displayed.

3. Click **Access Revoke** next to the required application.

   The **Access Revoke** tab of the **Share Settings** screen is displayed. The list of users or teams with whom the application has been shared is displayed in this section.

4. Select the user or team from which you want to revoke the access.

5. Click **Save**.

   The access to the selected application is revoked. An email is sent to the users or teams to notify them the revoke of their access.

The users or teams can not access the application anymore.

10 Managing API Packages and Plans

API Packages and Plans

API Package refers to a logical grouping of multiple APIs from a single API provider. A package can contain one or more APIs and an API can belong to more than one package. You should have the API Administrator or Provider privileges to manage an API Package. As a consumer you can only view the API package details. Once a package is deprecated, it is no longer visible to an API Consumer.

An API Plan is the contract proposal presented to consumers who are about to subscribe APIs. Plans are offered as tiered offerings with varying availability guarantees, SLAs or cost structures associated to it. An API Package can be associated with multiple plans at a time. This helps the API Providers in providing tiered access to their APIs to allow for different service levels and pricing plans. Though you can edit or delete a plan that has subscribers Software AG recommends you not to do so.

You can create packages and plans, associate a plan with a package, and associate APIs with a package through REST services. You can view the list of packages, package details, and APIs and plans associated with the package in the API Portal User Interface.

Creating an API Package

You can create an API Package using a REST service. The following section illustrates the structures for the REST service to create an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages` .

To create an API Package using REST service

- HTTP Method: POST

  Parameters: NA

  Request Payload

```
{
    "name":"Money conversion Package",
    "description":"Packages for money conversion",
    "icon":"http://www.digitalchalk.com/wp-content/uploads/2014/06/
     ecommerce-pricing-icon1.png"
}
```

  Response Payload

```
{
  "id": "f212c521-5334-11e6-3564-0050569d1bdd",
  "name": "Money conversion Package",
  "_link": "/abs/apirepository/v1/packages/f212c521-5334-11e6-3564-0050569d1bdd"
}
```

Creating a Plan

You can create a plan using a REST service. The following section illustrates the structures for the REST service to create a plan `http://<host>:<portnumber>/abs/apirepository/v1/plans` .

To create a plan using REST service

- HTTP Method: POST

  Parameters: NA

  Request Payload

```
{
 "name": "Blue Diamond Offer",
 "description": "Blue Diamond Offer 1 Description",
 "icon": "https://d30y9cdsu7xlg0.cloudfront.net/png/18095-200.png",
 "deprecated": false,
 "cost": {
  "currency": null,
  "value": "1000",
  "duration": null
 },
 "terms": "Blue Diamond 1 Terms",
 "license": "Blue Diamond 1",
 "security": [
  "APIKey"
 ],
 "enforcements": [{
  "name": "EnforcementA1",
  "properties": [{
   "key": "softlimit",
   "value": "value11"
  }, {
   "key": "hardlimit",
   "value": "value12"
  }]
 }, {
  "name": "EnforcementA2",
  "properties": [{
   "key": "days",
   "value": "value2"
  }]
 }]
}
```

  Response Payload

```
{
  "id": "69ce6721-5337-11e6-3564-0050569d1bdd",
  "name": "SAG_Plan Offer",
  "_link": "/abs/apirepository/v1/plans/69ce6721-5337-11e6-3564-0050569d1bdd"
}
```

Associating a Plan to a Package

You can associate plans with an API Package using a REST service. The following section illustrates the structures for the REST service to link a Plan with an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/plans` .

To associate a plan to a package

- HTTP Method: PUT

  Parameters: NA

  Request Payload

```
["6306b200-338c-11e6-61d0-d4bed967adb3",
     "16efea16-753b-4d7a-b5b7-ac0486934e31"]
```

  Response Payload

```
{
    "status": "PLANS_LINK_SUCCESSFUL",
  "message": "Plan(s) linked Successfully with {packagename}"
}
```

Disassociating a Plan from an API Package

You can disassociate a plan from an API Package using a REST service. The following section illustrates the structures for the REST service to unlink a plan from an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/plans` .

To unlink a plan from an API Package

- HTTP Method: DELETE

  Parameters: NA

  Request Payload

```
["6306b200-338c-11e6-61d0-d4bed967adb3",
          "16efea16-753b-4d7a-b5b7-ac0486934e31"]
```

  Response Payload

```
{
    "status": "PLANS_UNLINK_SUCCESSFUL",
  "message": "Plan(s) unlinked Successfully from {packagename}"
}
```

Deleting an API Package

You can delete an API Package using a REST service. The following section illustrates the structures for the REST service to delete an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}` .

To delete an API Package

- HTTP Method: DELETE

  Parameters: NA

Associating APIs with a Package

You can associate APIs with a Package using a REST service. The following section illustrates the structures for the REST service to link APIs with a Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/apis` .

To associate APIs with a package

- HTTP Method: PUT

  Parameters: NA

  Request Payload

```
["6306b200-338c-11e6-61d0-d4bed967adb3",
     "16efea16-753b-4d7a-b5b7-ac0486934e31"]
```

  Response Payload

```
{
  "status": "APIS_LINK_SUCCESSFUL",
  "message": "API(s) linked Successfully with {packagename}"
}
```

Disassociating APIs from an API Package

You can disassociate APIs from an API Package using a REST service. The following section illustrates the structures for the REST service to unlink APIs from an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/apis` .

To disassociate APIs from a package

- HTTP Method: DELETE

  Parameters: NA

  Request Payload

```
["6306b200-338c-11e6-61d0-d4bed967adb3",
```

```
          "16efea16-753b-4d7a-b5b7-ac0486934e31"]
```

Response Payload

```
{
  "status": "APIS_UNLINK_SUCCESSFUL",
  "message": "API(s) unlinked Successfully from {packagename}"
}
```

Associating Providers with a Package

You can associate a Provider with a Package using a REST service. The following section illustrates the structures for the REST service to link providers to a package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/providers?providerId=<value>`.

To associate providers with a package

- HTTP Method: PUT

  Parameters: NA

  Request Payload : NA

  Response Payload

```
{
 "status": "LINK_SUCCESSFUL",
 "message": "link successful"
}
```

Disassociating Providers from an API Package

You can disassociate providers from an API Package using a REST service. The following section illustrates the structures for the REST service to unlink providers from an API Package `http://<host>:<portnumber>/abs/apirepository/v1/packages/{id}/providers?providerId=<value>`.

To disassociate providers from an API package

- HTTP Method: DELETE

  Parameters: NA

  Request Payload: NA

  Response Payload

```
{
 "status": "UNLINK_SUCCESSFUL",
 "message": "unlink successful"
}
```

Viewing API Packages and Associated Plans

You can view the API Packages and the associated plans in the Packages page.

To view API Packages

1. Log in to API Portal.
2. Click **API Packages** in the title bar.

   A list of API packages is displayed. Additional information on the number of APIs and Plans associated with the package are also present.
3. Click **View details** for an API Package.

   A list of APIs and Plans associated with the API Package are displayed. You can click an API to view details of the associated API. Click **Subscribe** to subscribe a Plan. This opens Request Access Token dialog where you can request for one and subscribe to the plan.

   Note:
   You cannot filter APIs based on their stages.

API Package details

You can view the associated APIs and Plans from the Package details page. You can view details of a Package, details of the API, and subscribe for a plan from the API Package details view.

Field	Description
About	A brief description of the API Package
APIs	A list of APIs associated with the API Package. Click on individual APIs to view details of the API.
Plans	List of Plans associated with the API Package. Click **Subscribe** to request access token and subscribe to the particular Plan.

11 Managing Apps

App Gallery

The App gallery in API Portal acts as a market place in which API Portal users post their solution built on top of APIs published in the Portal; the market place can be used for establishing ecosystems for engaging partners and customers. Users who create these solutions can document them by providing descriptions and linking to other documents, features and screen shots of the app, and provide a link to external sources such as Apple Store or Google Play store. Other users can access the solutions available in the market place and download any assets posted in them.

The App gallery lists all the Apps that are present. Click an App to view its details.

App Details View

The App details view displays the details of the selected App such as App description, App features, and additional information about the App. You can also look at the discussion on the support forum regarding this App, choose to follow this App, rate the App or view the list of followers of the selected App.

Field	Description
Features	Displays a list of features available in this app. The features are categorised as the basic features and the advanced features that can be availed with an upgrade of the app.
Additional information	Displays additional information about the app as to when it was created, when it was last updated, the version of the app and the organization it belongs to.
Developed using	Lists the APIs that were used to build this app. Click on an API in the list to view the API details.
Screenshots	Displays the screen shots of the App.
Comments	Displays the comments that were added by the users. The comment text box allows a user to include a comment for the App. You can attach a tag, link or a file as part of the comment. In addition you can bookmark a comment, edit it, tag it, flag it, or delete it.
Developed by	Displays the names of the developers who developed the app.
Support forum	Click this option to navigate you to the Collaboration > My feed view where you can view the recent posts or activity. You can also post it to your feeds.
Follow this App	Click to follow this App.

Field	Description
	If you are already following this App you see the option Unfollow this App, in which case you can click it to unfollow the App.
List of followers	Displays the number of followers who are following this App.
Rate this App	Click the number of stars depending on how you want to rate this App.

Manage Apps

You can manage Apps from the Manage apps page. The page lists all the Apps and their description. You can edit an App, or delete an App from this view.

Field	Description
Name	The name of the App.
Description	The description about the App.
Version	The version number of the App.
Actions	Actions that can be performed on the App. Available options are: ■ Delete ■ Edit . This option is available only if you are the creator of the app.

Creating an App

You must have the privileges of an API Portal Administrator, an API Provider or a Consumer to create Apps.

To create an App

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage apps**.
3. Click **Create** in the Manage apps page.
4. In the Overview section, provide the following information:

a. In the **Name** field, provide a name for the app to identify it.

b. In the **Description** field, provide a short description about the app.

c. In the **Features** field, elaborate the features of the app.

d. In the **Icon** field, attach a image that is displayed as an icon for the app in the app gallery. Click Browse through the available icons and upload it.

> **Note:**
> The format and size is as mentioned in the configuration settings page.

e. In the **Version** field, provide a version for the app. If no version is mentioned, it automatically gives the version number 1.0.

f. In the **Company** field, provide the name of the company developing the app.

g. In the **Tags** field, provide tags for the app.

Tags are used as keywords to denote the app, such as an app predicting the weather could have tags like weather, and so on.

5. In the APIs section, perform the following to associate APIs to the app:

a. Click **+ Add**.

b. Select the required APIs

c. Click **OK**.

The selected APIs are displayed as a list. If you want to remove the APIs, click **Unlink** for the corresponding API and it gets deleted from the App.

6. In the Screenshot section, do the following:

a. Click **+ Add**.

b. Select the required image.

c. Click **Open**.

The screenshot is uploaded and is listed. The format and size is as mentioned in the configuration settings page.

7. In the Platforms section, type the URL to your App available in the respective platforms.

The URL provided is that of your app in the App Store, like https://play.google.com/store/apps/details?id=com.whatsapp&hl=en

8. Click **Save**.

   The App is created and listed in the Manage apps page.

Deleting an App

You must have the privileges of an API Portal Administrator or you must be the creator of the App to delete an App.

To delete an App

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage apps**.
3. Click corresponding to the App that is to be deleted.
4. Click **Yes** in the confirmation dialog.

Modifying Details of an App

You must be the creator of the App to modify its details.

To modify details of an App

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Manage apps**.
3. Click the corresponding to the App whose details are to be modified.
4. Modify the required details.
5. Click **Save**.

   The App is updated with the modified information and listed in the Manage apps page.

12 Managing Communities

Communities

Communities in API Portal define the exposure of APIs and packages to consumers. A community can have one or more administrators that manage the members of a community. The APIs and packages assigned to the public community are exposed to all users including unregistered users.

There are two types of communities - Private Communities and Public Communities.

API Administrators have the privileges to assign community administrators, add users to, and remove users from a community. APIs are assigned to specific communities by API Providers and the packages are published to communities from API Gateway. API Consumers have access to APIs and packages depending on whether they belong to a specific community or not. Consumers can view the API(s) that belong to a community by grouping the APIs by Communities in the API Gallery page.

Private Community

A *private community* is a group of users in API Portal that users can join through an invitation. Community membership grants API consumers access to private APIs and packages. APIs and packages that are associated to a private community are only visible to users that belong to this community. Each private community has one or more community administrators. Community administrators can add users to or remove users from the community.

To add existing users, the administrator needs the full user ID.

New users are invited to join API Portal and the community of the inviter. New users invited to a private community are not added to the public community.

If users do not belong to any other community including public communities, they are removed from API Portal; and their access tokens are revoked.

Note:
If you want to add an API or a package to a private community and restrict its visibility to the community members, you have to add the API or package to the private community and then remove it from the public community.

Public Community

The *public community* is a community any registered user can join. Each API Portal instance features only one public community. By default, all APIs and packages are assigned to the Public Community and they are visible to all users. This is the default community of a user joining without specifying a private community. Public community members and guest users can see the public APIs and packages.

Communities View

You can view the list of communities you are a part of in this view. You can click the  in the right top corner of the API Portal window and click Communities to navigate to this view. You can manage the communities if you are a Portal Administrator or if you are a community

administrator. As a Portal Administrator you can create a new community, add members to a community, define a community administrator, add or remove APIs from a community, add or remove user groups from a community, and so on.

Field	Description
Create	Click to create a new community.
Delete	Select a community and click this option to delete a community.
Name	Specifies the name of the community
Description	Specifies the description of the community.

Creating a Community

Prerequisite

You must have the privileges of a **Portal Administrator**.

To create a community:

1. Click in the right top corner of API Portal window to display the menu options.
2. Click **Communities**.
3. Click **Create**.
4. In the Create community page, type the name and description of the community.
5. Add members to the community as follows:
   - To add an unregistered user, type the e-mail address of the new user and click Add. An email is sent to the user.
   - To add an already existing user, click Add. The Select user(s) dialog box opens.
     1. To add existing users, select the relevant users and click **Add**.

        To add all users displayed click **Add all**.
     2. Click **OK**.

        The members added are listed under Members in the Create community page.
6. Add User groups to the community as follows:

   a. Click Add.

The Select user group(s) dialog box opens. This lists all the available LDAP user groups that are synchronized in the UMC and can be associated with the community. For more details on synchronizing users and user groups in UMC, see "Synchronizing LDAP Users or User Groups with User Management Console" on page 108

b. Select the required user groups and click **Add**.

To add all user groups displayed click **Add all**.

c. Click **OK**.

The user groups added are listed under User groups in the Create community page.

7. To add APIs to the community, click 

.

a. In the Add APIs to community dialog box, select the required APIs.

b. Click **OK**.

The APIs added are listed under APIs in the Create community page

8. Click **Apply**.

You created a community. This community appears in the list of communities displayed in the Communities page.

After creating a private community containing specific APIs and users, run the following command in ACC console, to make the API documents in the API details page, inaccessible to guest users.

```
reconfigure apiportalbundle_runnable +JAVA-Dcom.aris.ads.anonymous.access.allowed=false
```

When a guest user tries to access these documents, the user is prompted to provide user credentials.

Editing Community Details

Prerequisite

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To edit the details of a community:

1. Click  in the right top corner of API Portal window to display the menu options.

2. Click **Communities**.

The list of communities is displayed.

3. Click the name of the community that you want to edit.

The following details of the selected community are displayed.

- **Overview** - Displays the name and description of the community.
- **Members** - Displays the list of users, and their e-mail addresses; also indicates if there is any Administrator from them.
- **User Groups** - Displays the list of user groups.
- **APIs** - Displays the list of APIs that are published to the community.
- **Packages** - Displays the list of API Packages published to the community.

4. Edit the required details. For steps to edit the details, see "Creating a Community" on page 185.

5. Click **Apply**.

Your changes are saved.

Note:
You cannot add or remove the packages published to a community.

Defining Community Administrator

You can define a community administrator for a community.

Prerequisite

You are member of the **API Administrator** user group or you are administrator of the given community.

To define a community administrator:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name to which an administrator should be added.
4. Click the **Edit** button.
5. Click **Members**.
6. Select the **Administrator** column to add a user to the community's administrators.
7. Click **Apply**.

You defined an administrator for the community.

Adding Members to a Community

As an administrator of a community you can add members to the community.

Prerequisite

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To add members to a community:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name to which users should be added.
4. Click **Edit**.
5. Click **Members**.
6. Add members to the community as follows:
   - To add an unregistered user, type the e-mail address of the new user and click Add. An email is sent to the user.
   - To add an already existing user, click Add. The Select user(s) dialog box opens.
     1. To add existing users, select the relevant users and click **Add**.

        To add all users displayed click **Add all**.
     2. Click **OK**.

        The members added are listed under Members in the Edit community page.
7. Select the **Administrator** column to add a user to the community's administrators.
8. Click **Apply**.

You added users to the community.

Removing Members from a Community

You can remove members from a community.

Prerequisite

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To remove members from a community:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name from which users should be removed.
4. Click **Edit**.
5. Click **Members**.
6. To remove a user, click Delete in the user's row.
7. Click **Apply**.

You removed users from the community.

Adding User Groups to a Community

As an administrator of a community you can add user groups to the community. These are the LDAP user groups

Prerequisite

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To add members to a community:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name to which user groups are to be added.
4. Click **Edit**.
5. Click **User groups**.
6. Add User groups to the community as follows:
   a. Click Add. The Select user group(s) dialog box opens.

b. Select the relevant user groups and click **Add**.

The user groups that are imported into UMC are visible in the list of available user groups. To add all user groups displayed click **Add all**.

c. Click **OK**.

The user groups added are listed under User groups in the Create community page.

7. Click **Apply**.

You added user groups to the community. The added LDAP group user can view APIs which are admissible only to the community to which the user belongs.

Removing User Groups from a Community

You can remove user groups from a community.

Prerequisite

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To remove members from a community:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name from which the User groups should be removed.
4. Click **Edit**.
5. Click **User groups**.
6. Click Delete in the user group's row.
7. Click **Apply**.

You removed user groups from the community.

Removing a User from the Community Administrator Group

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To remove user from the community administrator group:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name you want to modify.
4. Click **Edit**.
5. Click **Members**.
6. Clear the **Administrator** column to remove the user from the community administrator group.
7. Click **Apply**.

You removed a user from the community administrator group.

Leaving a Community

You have to be a member of a community to leave it.

To leave a community

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community you want to leave.
4. Click **Leave**. A confirmation dialog is displayed.
5. Click **Leave**.

You left the community. If you left the last community you will be off-boarded from the API Portal.

Note:
If a user leaves a community, the API Gallery page would take couple of minutes to synchronize the changes.

Adding APIs to a Community

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

To add APIs to a community

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name to which APIs should be added.
4. Click **Edit**.
5. Click **APIs**.
6. Click **Add**. The Add APIs dialog opens.
7. Select the APIs to be added to this community and click **OK**.
8. Click **Apply**.

You added APIs to the community.

Removing APIs from a Community

You must have the privileges of a **Portal Administrator** or you must be the administrator of the given community.

You can remove the unwanted APIs from a community.

To remove APIs from a community

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Communities**.
3. Click the community name from which the APIs should be removed.
4. Click **Edit**.
5. Click **APIs**.
6. Click **Delete** in the API's row.
7. Click **Apply**.

You removed APIs from the community.

13 Managing Collaboration

Collaboration

Collaboration is the platform for cooperation across teams. With Collaboration information can be exchanged faster, knowledge can be shared and cooperation across borders is improved.

Group Coordinators and Collaboration Administrators manage the activities under Collaboration.

A Group Coordinator manages the group profile, edits requests to join private groups, privileges, and facilitates access and feed activity. Group Coordinators can delete their groups. As the creator of a group you are automatically the coordinator. You can grant user administrator privileges or revoke them from the members.

A Collaboration administrator has the same privileges as a coordinator. In addition, the Collaboration Administrator manages the posts that users have flagged, view all posts, delete them, and check activities reported as inappropriate.

Groups can be created for teams, departments, interest groups, topics, projects, and so on. There are public and private groups. Public groups are open to all users. For private groups, a coordinator decides who is to be granted access privileges to the group.

These are the five steps that are an optimal start to Collaboration:

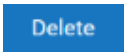
1. "Modifying a User Profile" on page 196
2. "Finding Users and Groups and Following their Feeds" on page 197
3. "Defining Filters" on page 198
4. "Commenting on, Sharing, and Flagging Posts" on page 199
5. "Inviting other Users to Collaboration" on page 200

Collaboration View

You can manage the activities from the Collaboration view if you are a Coordinator or a Collaboration Administrator. You can click the  in the right top corner of the API Portal window and click Collaboration to navigate to this view.

From the collaboration view you can manage the group profile, edit requests to join private groups, privileges, facilitate access and feed activity, manage the posts that users have flagged, view all posts, delete them, and check activities reported as inappropriate. Various sections in this view and what operations you can perform from each of them is described in the table.

Field	Description
	Specifies the number of new notifications. Click to view the notifications. If there are more than 3 notifications the Notifications dialog opens.

Field	Description
	You can click a notification message to see the details. Once you select a notification it is marked as read and the number of new notifications changes. Click **Unread** to see a list of all unread notifications. Click **Read**. A list of read notifications is displayed. You can click individual notifications or you can click **Delete all** to delete all the read messages. Click **Mark all as read** to mark all the unread messages as read. Click **Change your notifications changes** to change the settings to define how the notifications are displayed.
<user>	Specifies the user name with which you have logged in. It displays Administrator System if you have logged in as a Portal Administrator. Click to view the <user> profile view This view lists the posts that were published and categorised as Recent, Popular, and Active. You can search the required post by typing the characters in the search box. Click  to edit the user profile settings. Click **Followers** to view the users that are following you Click **Following** to list the users you are following.
My feed	Lists all my feed categorised as Recent, Popular, and Active. You can search the required post by typing the characters in the search box.
All company feeds	Lists all the company feeds categorised as Recent, Popular, and Active. You can search the required post by typing the characters in the search box.
My portal feed	Lists all feed from my portal categorised as Recent, Popular, and Active. You can search the required post by typing the characters in the search box.
Administration	You can manage your portal feeds from here. **Check flagged activities** - displays a list of flagged content. **Manage portal feeds** - displays a list of portal feeds alphabetically sorted. You can modify or delete a feed. Clean up activities- Select a date and click  to clear all activities that are older than the date specified.

Field	Description
	Sync - Click **Sync** to synchronize the permissions of Portal items.
Find groups	You can search a group from All Groups or My Groups section. **Statistics** - Displays the most active feed data. **All groups** - You can search a group from All Groups or My Groups section. ■ Click **Create group** to create a new group. ■ Click **Follow** to follow a particular group. **My groups** - Lists the group created by you or where you are a member of. Click **Create group** to create a new group.
Create group	Click to create a new group.
My bookmarks	Lists all the bookmarks you have saved.
My liked items	Lists all the items you have liked.
Create filter	Click to create custom filters to find interesting posts quickly and easily using keywords or to gain a better overview.
Check flagged activities	Click to check the flagged activities from the list.
Manage portal feeds	Click to manage the portal feeds. You can also synchronize access privileges.
Manage announcements	Click to manage announcements
Configure document user	Click to configure the ARIS document storage user
Dashboards	Click to view the dashboards.
Cleanup and export	
Configure notifications	Click to configure notification and customize them

Modifying a User Profile

Modify your user profile to provide other users with information about your areas of activity and interests.

To modify a user profile:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.

3. Click *<user name>*.
4. Click **Edit profile**. The Edit profile page is displayed.
5. Upload a photo in JPG, PNG, or GIF format, with a maximum size of 4 MB. A square photo fits best.
6. After uploading, you can select the picture detail that you want to use. To do this, click **Edit**.
7. Use the frame to select the required picture detail.
8. Click **OK**.
9. In the **Title** field, provide your title that describes your position in the company.
10. In the **Description** field, provide a brief description of your job responsibilities.
11. In the **Keywords** field, specify keywords that will enable colleagues looking for particular information or expertise to find you. Use a comma as the separator.
12. In the **Phone** field, type your telephone number.
13. Select **Allow others to post to my feed** if you want your colleagues to be able to post information on your feed. and submit comments in your feeds.
14. Select **Allow others to comment on activities in my feed** if you want your colleagues to be able to submit comments in your feeds.
15. Click **OK**.

Your profile is now modified accordingly. The information is now available to other users.

Finding Users and Groups and Following their Feeds

You can look for colleagues or groups to find interesting contacts and information and follow their feeds.

To find users and groups and follow their feeds:

1. Type the name of the user or the group in the global find box at the top right. The search results are displayed dynamically. Continue typing characters until the relevant user or group is displayed.
2. Click the name you are looking for.

   The profile of the user or the group is displayed with all posts.

3. Do one of the following:

   - For users and public groups, click **Follow**.
   - For private groups, click **Send request**.

     Private groups are marked with a padlock icon.

When you follow users, you have access to the posts that they publish in their feeds. In private groups, a coordinator needs to confirm your request before you are allowed access to this group's posts, comments, and so on. In public groups you have immediate access to the group. The feeds you follow are displayed under **My feed** and the groups you follow under **Groups**. To stop following a user or group, click **Unfollow** in the user or group profile.

Users who are following your feeds are displayed under **Followers** in your profile.

Tip:

- To find groups, you can also use the group search function (**Groups > Find groups**). Alternatively, you can display the profile of another user to find out which groups that user belongs to.
- You can also use the global search to search for keywords, in order to find interesting feeds, users, and groups.

Defining Filters

You can define custom filters to find interesting posts quickly and easily using keywords or to gain a better overview.

To create a filter

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.
3. Click **Create filter**. The Create/edit filter page opens.
4. Select the relevant filter criteria, for example, whether you want all feeds or only feeds you follow to be included in the filter.
5. Type a name for the filter.
6. Specify the keywords that can be used to find relevant posts. Use a space to separate the keywords.
7. Click **OK**.

The filter is saved and is displayed under **Filters**. Click the filter name to display the posts that contain the specified keywords.

To change a saved filter, click **Edit filter**. To delete filters that you no longer require, click **Delete filter**.

Commenting on, Sharing, and Flagging Posts

Depending on whether a post is your own or from another user, you can perform different actions.

To comment on, share and flag posts

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.
3. Open the page containing the relevant post.
4. You can do one of the following:
   - Click **Like** to show other users what interests you. The user who wrote the post receives a notification, if they have made the relevant notification setting. The post is also flagged accordingly and added to your filter. To cancel your Like, click **Unlike**.
   - Click **Comment** to type a comment or to add additional information for a post. You can also add a link to a Web site.
   - Click **Share** to comment on a post by another user and publish it in your own feeds.

     To remove the post from your feeds, click **Delete**.
   - Click **Bookmark** to add a post to your filters so that you can easily find it again later (**Filter > My bookmarks**).
   - Click **Tag** to tag the post or a comment.
   - Click **Delete** to delete a post or a comment.
   - Click **Flag** to send a post to the Collaboration Administrator for review because you think it is inappropriate.
   - Click the timestamp, for example, **2 hours ago** to display a post with all un-abbreviated comments, and so on.

Depending on the selected action, the affected users receive a message by email.

Creating a Group

Create your own group if you cannot find any interesting groups or you need a group for your team. Groups enable users to collaborate in a team and to participate in a special interest group or a particular topic.

To create a group

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.
3. Click **Create group**.

   The Create group page opens.
4. In the **Name** field, type a name for the group.
5. In the **Short name** field, type a short name you would want the group to be displayed as.
6. In the **Description** field, type information that describes the purpose of the groups feed.
7. In the **Keywords** field, type keywords that describe your group so that the group is easily found in search results. You can type in multiple keywords separated by comma.
8. In the **Web address** field, type the URL of the website if your group has one.
9. Click **Add coordinator** to add an additional coordinator, if required, who will manage the group profile and privileges. You can type in the search string for the user you are trying to find to add as a coordinator in the Search boxProvide As the creator of the group, you are automatically the coordinator.

   a. In the Search dialog that appears, type in the search string for the user you are trying to find to add as a coordinator
10. Enable the relevant privacy option.
11. Click **OK**.

Your group is created. It is displayed under **Groups**. Using tags, other users can find the group and follow its posts. In private groups, only members are able to read the posts. The group's name and description will, however, be visible in search results for non-members, as well.

Inviting other Users to Collaboration

You can invite other users to become a member in a specific group.

To invite other users to a group:

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.

The Collaboration page appears.

3. Click the group to which you want to issue an invitation for.

4. In the **Add colleagues** box, enter the name of the person you want to invite.

5. Click **Add**.

The user receives a notification and can log in to Collaboration. Initially only read privileges are assigned, until the user becomes a member of the group .

Group coordinators can add more users to a group directly (**Add colleagues**). That user immediately becomes a member of the group and receives the posts, and comments by the group.

Accepting or Denying Requests to join Private Groups

You can accept or deny requests to join private groups of which you are a coordinator.

To accept or deny a request to join private groups:

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **Collaboration**.

3. Click the private group.

4. Click **Requests**. The user requests are displayed.

5. Click **Accept** to accept the request.

   You can click **Deny** to deny the request.

If the user was accepted as a group member the user will be notified accordingly and displayed in the list of followers. If the user was denied membership the user and all other group coordinators receive a corresponding notification.

Granting or Revoking Group Coordinator Privileges

You can grant group coordinator privileges to additional group members or revoke them from the members on a need basis.

To grant or revoke group administrator privileges:

1. Click in the right top corner of the API Portal window to display the menu options.

2. Click **Collaboration**.

3. Open a private group.

4. Click **Followers**. The members of this group are displayed.

5. In the row of the group member you can grant or revoke the administrator privileges by clicking **Grant** or **Revoke**.

The user is granted privileges or they are withdrawn. The user receives a notification. If a user is granted coordinator privileges all other coordinators of the group receive a notification, as well.

A coordinator can cancel group membership only if another coordinator withdraws the coordinator privileges from the user first.

Checking Activities Reported as Inappropriate

As a Collaboration Administrator, you can check content reported as inappropriate and decide if the content needs to be deleted.

To check activities reported as inappropriate

1. Open the notification you received for activities reported as inappropriate.

2. Click the link. The content reported is shown.

3. Verify whether the content violates the Collaboration terms of use.

4. Do one of the following:

   - Click **Approve** if the content is acceptable.
   - Click **Delete** if the content violates the terms of use.

The content reported was checked and depending on the result it is deleted or continues to be shown.

Modifying Notification Settings

You can specify the situations when you want to receive a notification about activities in API Portal and modify the settings accordingly.

To modify notification settings

1. Click 4 on the top left corner of the collaboration page to view notifications. The **Notifications** dialog is displayed.

2. Click **Change your notification settings**. The notification settings are displayed.

3. Specify the situations when you want to be informed about activities by other users or activities on groups. In each case, decide whether you want to receive the notification as an internal notification in API Portal(internal) or as an **e-mail**.

4. Click **Save**.

Your settings are saved. In the future, you will be notified in the selected situations.

Commenting on Portal Content

You can add comments to APIs and post information that could be of interest to your colleagues.

To comment on portal content

1. Click **API Gallery**.

2. Click **View details** for an API.

3. Click **Latest Posts**.

4. Type a comment. The text may have up to 250 characters.

   Note:
   If you want to add a link in the comment, the characters in the link are counted towards the 250 available characters.

5. Add links if required, by typing the required link and click **Add link**.

   The link is checked and added.

   If you add a link to a Web site , the link must start with http://.

6. Add tags, if required, by typing the required tags in the tag field.

7. Add a document, if required, by clicking  .

8. Click **Post**.

Your comment is posted.

Publishing Posts

You can post information that could be of interest to your colleagues, or start a discussion on a particular topic.

To publish a post

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.
3. Click **My feed**.
4. Type or copy the text into the input field. Up to 2000 characters are available.

   If you want to add a link the characters in the link are counted. To create a link to another user in the text, type '@' at the relevant point in the text, immediately followed by the name of the user.
5. You can attach a Tag, Link, and a File while publishing a post as follows:
   - To attach a Tag:
     1. Click <tag>.
     2. Add the tag.
     3. Press **Enter**.
   - To attach a Link:
     1. Click <link>.
     2. Add the Link. The link must start with http://
     3. Press **Enter**.
   - To attach a File:
     1. Click <File>.
     2. In the Select Document dialog, select a file by browsing in the left hand panel and click **OK**.

        Alternatively you can also search for the file by clicking search and typing the required text in the search box.
6. Click **Post**.

Your post is published. If you have published something in your feed, the text is also displayed in the company feeds. If you have posted something in another user's feed, this information is also displayed in your feed and in the company feed, and is indicated by **<user name A> for <user name B>**. If you published a post in a private group you are a member of, the post is shown only to group members.

Using Hashtags

You can use hashtags to categorize posts using keywords or topics. This enables other users to find posts on interesting topics more easily. A hashtag consists of the # character followed by a

keyword or phrase. There is no space after the # and the phrase does not contain any punctuation marks.

Examples

#BPM

#Optimize your processes using social collaboration

To use a hashtag

1. Type your post or comment in the relevant feed (your feeds, company feeds, group feeds).
2. Type the word to be used as a keyword, preceded by a # symbol, for example, #BPM, or enter a sentence. Alternatively, select an existing hashtag from the list.
3. Publish the post.

Your post is published and the keyword entered as a hashtag is highlighted in color as a link. Additionally, a tag is automatically created for the hashtag.

If a user clicks the hashtag, all post and comments subsequently entered are displayed on a separate page. A hashtag can be saved as a filter. You can then use the same hashtag in the filter definition, for example, #BPM.

Following API Portal content as a Group

You can follow any interesting content in API Portal as a group. This enables the group, for example, a project team, to jointly discuss and edit relevant processes.

Prerequisite

- You must be the coordinator of the group.
- The item from API Portal must either be followed as a portal feed or have comments in Collaboration.

To follow API Portal content as a group

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Collaboration**.
3. Activate the group that has to follow portal feeds.
4. Click **Edit group**.
5. Click **Follow portal feeds**. The dialog opens.

6. Type a search term for finding the portal feed. Ensure that the spelling is correct.

7. Select the required portal feed in the search result.

8. Click **OK** in both the dialog and the group.

Using the **Following** button, all members of the group can now view the portal feed.

The portal feeds a group is following are displayed in the group under **Following**. Comments are shown directly in the group. To stop following a portal feed, the coordinator removes the feed from the list of portal feeds being followed (by clicking **Delete**).

Users of the group are notified when the portal feed changes or comments are added to it.

Finding Help

In addition to this online help, there are various ways you can find support in Collaboration.

To find help

- Publish a post containing your question and use the hashtag**#Help**.

  This keyword enables other users to find and reply to your question. Alternatively, contact your system administrator.

14 Analytics

Dashboard

Dashboard displays a variety of charts to provide an overview of API Portal and it's API usage. Dashboard can be accessed by clicking the **Dashboard** option in the user action menu. Select the type of dashboard in the dashboard drop down list to view the particular dashboard.

The different types of dashboards available are:

- **Global dashboard** - used to view KPIs based on API Portal page views and API views by users.
- **API audit log** - used to view the lifecycle and access token events for an API, monitor the subscriptions per package, and access token requests per API.
- **User audit log** - used to track total number of logins, active sessions, number of consumers, the success and failure of logins, user registrations, and user audit log.
- **Runtime dashboard** - used to study the API's invocations per user and its performance during runtime.
- **API trends dashboard** - used to study the API invocation trends by response time, success and failure rates.
- **Consumer dashboard** - used to track the total API requests over a period of time, requests over time per API, and API request log.

You can create time based filters to view the statistics in the selected time range.

Select...	To...
Quick Time	You can select a time range from a defined list. For example, **Last 24 hours**, **Previous week**, and so on. The data displayed would be the data captured in the time range specified. By default, the time range selected is **Last 24 hours**.
Relative Time	You can select a time relative to the current time. The range can be specified in minutes, hours, days, weeks, and months. For example: Selecting **10 days ago** in the **From** field and **Now** in the **To** field displays the data in the time interval chosen.
Absolute Time	You can specify an absolute time range, by selecting the required dates in the date picker. For example, selecting specific dates in the **From** and **To** fields displays data in the time interval chosen.

Note:
Localization is not supported in API Portal dashboards and you cannot filter APIs based on their stages

Global Dashboard

The Global Dashboard captures KPIs based on API Portal page views and API views by users. You can filter the data as required and view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
Page views	Tracks the page views by users (both registered users and guest users). This data provides an insight into which portal pages have the most or least number of views. For example, a large number of page views for an API's detail page could indicate that it is a popular API. Top 10 such page views are listed in the chart.
Page views (Guest vs Registered)	Tracks the page views by users who visit API Portal as guest users versus registered users.
Visits per API	Tracks the visits to the details page of a particular API. The top 10 APIs are displayed.
Unique visitors	Tracks the total number of unique visitors who have accessed API Portal during a specified time interval.
API views over time	Tracks the overall API views over time in the specified time interval.
Page views over time	Tracks the overall page views over time in the specified time interval.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

API Audit Log

The API audit log dashboard is used to view the lifecycle and access token events for an API, monitor the subscriptions per package, and access token requests per API. You can filter the data as required and view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
API life cycle events	Tracks events of an API like Publish, RePublish, and UnPublish over time. The different coloured lines in the line graph depict the different categories of events.
Access token events	Tracks all access token related activities like RequestAccessToken, RenewAccessToken, and RevokeAccessToken over time.
Access token event distribution	Displays the distribution of access token events like RequestAccessToken, RenewAccessToken, and RevokeAccessToken.
Access token requests per API	Tracks the number of access token requests for an API. Displays the top 5 APIs.
Subscriptions per package	Tracks the number of subscriptions per package. Each package can have multiple plans associated with it. So the subscriptions of a package are displayed as a stacked bar chart depicting the subscriptions of each associated plan. It displays the top 5 packages.
Audit log	The table lists information of all the events sorted by the descending order of creation date. Details include: ■ **Time** – time when the event was logged. ■ **eventType** - the type of event that was generated. ■ **userName** – user who initiated the event. ■ **apiName** – name of the API. ■ **apiVersion** – version of the API. ■ **applicationName** – name of the application for which the event was generated. ■ **domainName** – name of the package. ■ **planName** - name of the plan.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

User Audit Log

The user audit log dashboard is used to track total number of logins, active sessions, number of consumers, the success and failure of logins, user registrations, and user audit log. You can filter the data as required and view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
Total login	Tracks and displays the total number of logins that have taken place in API Portal in the specified interval.
Active sessions	Tracks and displays the total number of users who are currently logged on to API Portal.
Total consumers	Tracks and displays the total number of consumer users in API Portal. This data is based on the users with API Consumer role.
Login success/failure	Tracks the proportion of success to failure cases of user login events. The success and failure events are depicted in different colors. Hover the mouse over required section to view additional details.
Registrations over time	Tracks the total number of users who have signed up in API Portal over the specified time. It includes users on-boarded through API Portal and through social networking sites.
Sources of user registration	Tracks the number of users who have registered in API Portal using the portal's sign-up flow and the users who have registered through social networking sites like Facebook and Google plus.
User audit log	The table lists all the user related events sorted by the descending order of creation date. The table provides the following information: ■ **Time** – time at which the event occurred.

KPI	Purpose
	■ **Type** – type of event. ■ **parameters.sessionid** – session id of the user. ■ **client IP** – IP of the system in which the event took place. ■ **parameters.username** – username of the user who is actually involved in the event.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

Runtime Dashboard

The Runtime Dashboard is used to study the API's invocations per user and its performance during runtime. You can filter the data as required and view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
Top 10 APIs	Tracks the top 10 APIs based on the API requests.
Requests per user	Depicts the distribution of API requests across users.
Top APIs by consumption	Tracks the top 10 APIs based on API's consumption. The different colors in the chart represents different consumer application name.
Overall events	Depicts the total number of events that are generated for the API published in API Portal for the specified time interval. The data is displayed in the form of a donut chart. The chart is sliced based on the distribution of event types.
Requests over time by API	Tracks the number of API requests over time.
Runtime events	The table displays the information of all generated events, sorted by the descending order of creation date.

KPI	Purpose
	Details include: ■ **Time** – time when the event was logged. ■ **eventType** – type of event . ■ **apiName** – name of the API for which the event was triggered. ■ **apiVersion** - version of the API. ■ **operationName** – name of the operation that triggered the event. ■ **applicationName** - name of the application that generated the event. ■ **applicationIP** - IP address of the application. ■ **request** – description for the alert generated. ■ **response** – source that triggered the alert. ■ **responseCode** - response code for the event. ■ **providerTime** - time interval in milliseconds from when a request is forwarded to a native provider until a response is received from the provider. ■ **totalTime** - time interval in milliseconds from when a request is received by the virtual service runtime until the response is returned to the caller.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

API Trends Dashboard

The API trends dashboard is used to study the API invocation trends by response time, success and failure rates. You can filter the data as required and view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
API trends by response	Tracks and compares the total successful invocations with total failed invocations for APIs present in the API Portal in the specified interval. The failure and the successful events are depicted by different colors in the graph.
API trend by success	Tracks the invocation trends for top 5 APIs based on most successful invocation in API Portal in the specified interval. The invocation trends from 5 different APIs are depicted by different colors.
API trend by failure	Tracks the invocation trends for top 5 APIs based on most failed invocation in API Portal in the specified interval. The invocation trends from 5 different APIs are depicted by different colors.
Overall average response time	Tracks the overall average response time for all APIs in API Portal over time in the specified interval.
Average Response time by API	This chart depicts the average response time for top 5 APIs inAPI Portal over time in the specified interval.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

Consumer Dashboard

Consumer dashboard captures the KPIs specific to the visitors who access API Portal. It is used to track the total API requests over a period of time, requests over time per API, and API request log.

You can filter the data using the two filters:

- **Application filter** - The Applications drop-down list displays all the applications of the user. You can select any application and the dashboard displays data for the application selected.

- **API filter** - On selecting an application, the APIs corresponding to that application are displayed in the APIs drop-down list. If there are multiple APIs associated with an application (in case of subscription tokens), you can choose an API from the APIs list.

The data displayed are based on the selected application and API combination. All the charts in this dashboard represent the data of the logged in user. You can view the required data by hovering the cursor over the legend in each of the charts.

The KPIs can be monitored in each of the charts displayed.

KPI	Purpose
Total requests (Today)	Tracks the total number of API requests made by the logged in user, for the day. This is independent of the time chosen in the time filter.
Total requests (Last 7 days)	Tracks the total number of API requests made by the logged in user, in the last 7 days. This is independent of the time chosen in the time filter.
Total requests (Last one month)	Tracks the total number of API requests made by the logged in user, in the last month. This is independent of the time chosen in the time filter.
Total requests (Chosen interval)	Tracks the total number of API requests made by the logged in user, for the time interval chosen in the time filter.
Requests per API	Tracks the number of requests for each API. For a versioned API, this chart is displayed in a stacked bar format for different versions of that API.
Requests over time by API	Tracks the number of API requests over time.
Average response time	Depicts the average response time of the top 5 APIs over time in the specified time interval.
API request log	Displays the information of all the transactions of the logged in user, sorted in the descending order of creation date. The table displays information such as time when the event was created, API Name, API version, name of the application that generated the event, IP address of the consumer, request and response details.

You can use the following options to resize the view and view the legend of the required KPI.

Settings	Description
	Click to view the legend of the chart. Hovering the cursor over any of the elements in the legend highlights the section pertaining to the legend chosen.
⌟	Use this to resize the chart.

Viewing Dashboards

Dashboards display a variety of charts to provide an overview of the API usage.

Prerequisite

You must have the **API Administrator** role.

To view a dashboard

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Dashboard**.
3. Select the dashboard type.

The Dashboards page opens and different charts are displayed.

15 Managing Data in API Portal

Configuring Audit Logs

API Portal provides a level of auditing that allows an API Portal Administrator to see the state of the API Portal and get information about events that occur.

Various events that can be audited are:

- Login attempts
- User Deletion
- API lifecycle
- Access token lifecycle
- API Provider lifecycle
- Purge logs

In addition to monitoring events through various audit logs the API Portal administrator can configure the following as required on a tenant basis:

- Type of auditing events
- Purging of different types of logs

Different event types and their respective list properties that have to be configured to enable or disable the auditing of these logs is listed in the following table.

Event Type	List Property Value
Approvals	com.aris.umc.apiportal.useronboarding.approval.purge
Request Access Token	com.aris.apiportal.eventType.requestaccesstoken.log.enabled
Renew Access Token	com.aris.apiportal.eventType.renewaccesstoken.log.enabled
Revoke Access Token	com.aris.apiportal.eventType.revokeaccesstoken.log.enabled
Publish API	com.aris.apiportal.eventType.publish.log.enabled
Republish API	com.aris.apiportal.eventType.republish.log.enabled
Unpublish API	com.aris.apiportal.eventType.unpublish.log.enabled
Purge Log	com.aris.apiportal.eventType.purge.log.enabled

You can configure the Audit logs to enable or disable the required types of auditing events such as login events, API lifecycle events, access token lifecycle, user deletion, and purge logs. These are enabled by default.

You can configure audit logs by invoking the REST service.

Endpoint: `http://`<hostname>`:`<port>`/abs/apirepository/configurations/`<PropertyName>`/`

Supported HTTP methods:

- GET - Display the current audit setting (enabled/disabled) for the given property.
- POST - Modify the audit setting for the given property. Message body to be set to false (to disable logging).

For example, to disable the audit logging for the publish property type:

Endpoint: http://*<hostname>*:*<port>*/abs//apirepository/configurations/com.aris.apiportal.eventType.publish.log.enabled/

HTTP Method: POST

Message body: false

For example, to enable the audit logging for the publish property type:

Endpoint: `http://<hostname>:<port>/abs//apirepository/configurations/com.aris.apiportal.eventType.publish.log.enabled/`

HTTP Method: POST

Message body: true

For example, to get the status of the properties:

Endpoint URL:
`http://<hostname>:<port>/abs/apirepository/configurations/com.aris.apiportal.eventType.publish.log.enabled/`

HTTP Method: GET

Response:

```
{
propName: "com.aris.apiportal.eventType.publish.log.enabled"
propValue: "true"
}
```

Purging Logs

The process of systematically deleting unwanted data from the database is called purging. Each time you perform a purge operation, the corresponding purging log entry is created and can be viewed in API Audit log page in the API Portal User Interface. For example, when you purge a certain set of analytics logs, the corresponding purging log entry is created and is displayed in the API Audit log page in the API Portal User Interface. You can purge logs by setting the required date or duration per tenant in the API Portal.

You can purge logs in one of the following ways:

- "Purging logs by invoking a REST service" on page 220
- "Purging logs through the user interface" on page 220

The purge service supports the following types of log entries:

- Audit logs - API and User audit logs
- Analytics - Page Views and other analytics data
- Approval - Objects of approved users
- Stale Users - Users who have not been activated through the email confirmation workflow
- API analytics - Run-time analytics data for APIs

Purging logs by invoking a REST service

You can purge logs by invoking a REST service.

- Endpoint: http://<*hostname*>:<*port*>/abs/apirepository/purge/

  Method: DELETE

  Content-type: application/json

  Request payload

```
{
          "purgeSettings":[{"objectType":"Audit Logs",
                            "until":"2014-11-29T00:00:00"},
                           {"objectType":"Analytics",
                            "until":"2014-11-29T00:00:00"},
                           {"objectType":"Approval", "olderThan":"1m"},
                           {"objectType":"Stale Users", "olderThan":"1m"}]
}
```

Purging logs through the user interface

As an API Portal Administrator, you can purge the following types, Audit Logs, Analytics, API Analyitics, Approval and Stale users. You can apply filters using Date or Duration.

To purge logs from the API Portal user interface:

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Manage logs**.
4. Click **Purge logs**.
5. On the **Purge logs** page, do one of the following:
   - Select **Until** and select a date until which you want the data to be purged. The rest of the data is retained.

- Select **Duration** and type the duration value for which you want the data to be retained. The rest of the data is purged.

6. Select the type of data that need to be purged.

7. Click **Purge**. A confirmation message is displayed.

8. Click **Yes**.

The selected data is purged.

Backing up and Restoring Tenant-specific Data

This section covers information on how an API Portal Administrator can back up or restore tenant-specific data.

API Portal is multi-tenant capable and each tenant administrator can manage their data. Generally, the tenant-specific data comprises of

- **API Portal objects**: APIs, Access Keys, and other related data, users, user groups, licenses, and configurations, API documents and files.
- **Collaboration data**: Collaboration groups, posts and discussions related data, and comments or posts related to APIs.
- **API Portal analytics**: Page views, user registrations, API lifecycle events, access token lifecycle events, provider lifecycle events, and run-time analytics data for APIs.

You can back up and restore tenant-specific data in the following ways:

- Using the API Portal Cloud Controller (ACC)
- From the API Portal User Interface

Prerequisites for Backup and Restore in a Distributed Environment

The following points are to be considered if the API Portal server is installed in a clustered high availability setup.

Note:
If the API Portal is a single node installation, then you can skip this section.

- In a typical multi-node setup, the different parts of the application (runnables) are installed on the different nodes (machines). It is important to verify that all the nodes are up and accessible. All the nodes must be registered to the parent node and the API Portal Cloud Controller (ACC) needs to know about your distributed environment. There are a couple of ways to register the different nodes to ACC.

Registering Nodes Using the ACC REST Service

To register nodes using ACC REST service

- Post a http request to `http://loadbalancer_host:loadbalancer_port/acc/rest/nodes/` with payload:

```
{
 "nodename": "<<hostname>>",
"hostname": "<<hostname>>",
"port": "<<acc_portno>>",
"username": "Clous",
"password": "g3h31m"
}
```

  The user must be a member of infrastructure administrator tenant - the master tenant - with Tenant administrator functional privilege.

Your system is now ready for taking the backups or restoring the backups.

Registering Nodes Using the Configuration File

To register nodes using the configuration file

1. Stop the component apiportalbundle_s through ACC using the `stop apiportalbundle_s` command.

2. Edit the file in the path *Installation directory*\API_Portal\server\bin\work\apiportalbundle_s\base\webapps\abs\WEB-INF\config\spring\ext\apiportal-servlet.xml.

3. Add the new entry for the distributed ACC server host and port in the map.

```
<bean id="addNodeAction"
  class="com.aris.modeling.server.webapp.apiportal.startup.impl.AddNodeAction">
    <property name="nodeList">
        <map>
            <entry key="127.0.0.1" value="9001" />
            <entry key="localhost" value="18002" />
        </map>
    </property>
</bean>
```

4. Restart the component apiportalbundle_s through ACC using the `start apiportalbundle_s` command.

Your system is now ready for the taking the backups or restoring the backups.

Backing up Tenant-specific Data

The user must be a member of the API Administrator user group.

To backup tenant-specific data

1. Start the API Portal Cloud Controller (ACC).
2. Run the following command to back up the tenant-specific data:

```
backup tenant tenant-name for type1, type2
```

Note:
Select the following types depending on the data to be backed up:

- ABS, ADS, UMC to back up API Portal objects data.
- ECP to back up collaboration data.
- APIPORTAL to back up the lifecycle events and run-time analytics data.

For example, if you want to backup only the collaboration and API Portal analytics data from the default tenant the command format would be:

```
backup tenant default for ECP, APIPORTAL
```

Backing up Tenant-specific Data Through the User Interface

You can back up tenant-specific data that can be used when required. You can define the type of data to be backed up, for example, All Data, Collaboration data or API Portal objects.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To back up tenant data

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Administration** > **Manage data**.
3. Select **Backup**.
4. Select the relevant options. The available options are **All**, **API Portal objects**, **Collaboration data**, and **API Portal analytics**. You must select at least one option.
5. Click **Backup**.
6. Confirm your password.

A success message appears when the backup process is completed. The backup file with an extension `.acb` is created and saved in the downloads section. You can move the file and save it in another location of your choice.

Restoring Tenant-specific Data

The user must be a member of the API Administrator user group.

Note:
Restoring overwrites the existing content in your API Portal instance. Only the user metadata is merged.

When you restore tenant-specific data, the data that is restored depends on the data that is contained in the backup file and the types (all data, API Portal objects, Collaboration data or API analytics) you select.

To restore tenant-specific data

1. Start the API Portal Cloud Controller (ACC).

2. Run one of the following commands to back up the tenant-specific data:

   - To restore data of all app types:

     ```
     restore tenant tenant_name from backup_location
     username=admin_username password=admin_password
     ```

   - To restore data of a particular app type:

     ```
     restore tenant tenant_name for apptype1, apptype2 from
     backup_location username=admin_username password=admin_password
     ```

   Note:
   Select the following types depending on the data to be restored:
   - ABS, ADS, UMC to back up API Portal objects data.
   - ECP to back up collaboration data.
   - APIPORTAL to back up the lifecycle events and run-time analytics data.

   For example, if you want to restore only the collaboration and API Portal analytics data from the backed up file, mybackup.acb located in the C drive, on to the default tenant, the command format would be:

   ```
   restore tenant default for ECP, APIPORTAL from c:\mybackup.acb
   username=adminuser password=adminpass
   ```

Restoring Tenant-specific Data Through the User Interface

Depending on the data that is contained in the backup file, All data, API Portal objects, Collaboration data, or API Portal analytics data are restored.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To restore tenant-specific data

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration** > **Manage data**.
3. Select **Restore**.
4. Click **Upload** and select the relevant backup file to be uploaded.
5. Select the relevant options. The available options are **All**, **API Portal objects**, **Collaboration data**, and **API Portal analytics**. You must select at least one option.

   The data that is restored depends on the data that is contained in the backup file and the types (all data, API Portal objects, Collaboration data or API analytics) you select.
6. Click **Restore**.

   A success message appears when the restore process is completed.

Troubleshooting Backup and Restore Failures

In case the back up or restore process fails, check whether all components are running. To do so, use API Portal Cloud Controller (ACC).

To troubleshoot backup and restore failures

1. Check whether all components are running using ACC.

   a. To start ACC under a Windows operating system click **Start > All Programs > webMethods API Portal > Administration > Start webMethods API Portal Cloud Controller**.

      To start ACC under a Linux operating system, run the **acc.sh** file.

   b. Type **help** or **help <command>** to get information about the usage of the commands.

   c. Type **list** to monitor the status of all components (runnables). This example shows ACC of a API Portal Connect Server installation for a medium number of users.

      All Components with their states are listed.

```
ACC+ Node localhost - 7 installed runnables.
ID                State    Version    Type
```

```
zoo_s              STARTED  10.0.14.0  com.aris.runnables.zookeeper-run-prod
postgres_s         STARTED  10.0.14.0  com.aris.runnables.PostgreSQL-run-prod
cloudsearch_s      STARTED  10.0.14.0  com.aris.cip.y-cloudsearch-run-prod
elastic_s         STARTED  10.0.14.0  com.aris.runnables.elasticsearch-run-prod
kibana_s           STARTED  10.0.14.0  com.aris.runnables.kibana-run-prod
apiportalbundle_s  STARTED  10.0.14.0  com.aris.bundles.apiportalbundle-run-prod
loadbalancer_s     STARTED  10.0.14.0  com.aris.runnables.httpd.httpd-run-prod
```

The status of all components represented by their instance IDs are listed. Possible states are:

- **UNKNOWN**: The component state is not yet known. This state is shown directly after the agent was started.
- **STOPPED**: The component is currently not running.
- **STARTING**: The component is starting, but this process is not complete yet.
- **STARTED**: The component is running.
- **STOPPING**: The component is stopping, but this process is not complete yet.
- **DOWN**: This component has and crashed. The agent attempts to automatically restart the component momentarily.
- **FAILED**: Component has crashed. The agent has given up trying to restart the component.

Collecting API Portal Logs

As an API Portal Administrator, you can collect the logs that contain the API Portal data. You can collect all logs that are relevant for the API using the script collectlogfiles.

To collect API Portal logs

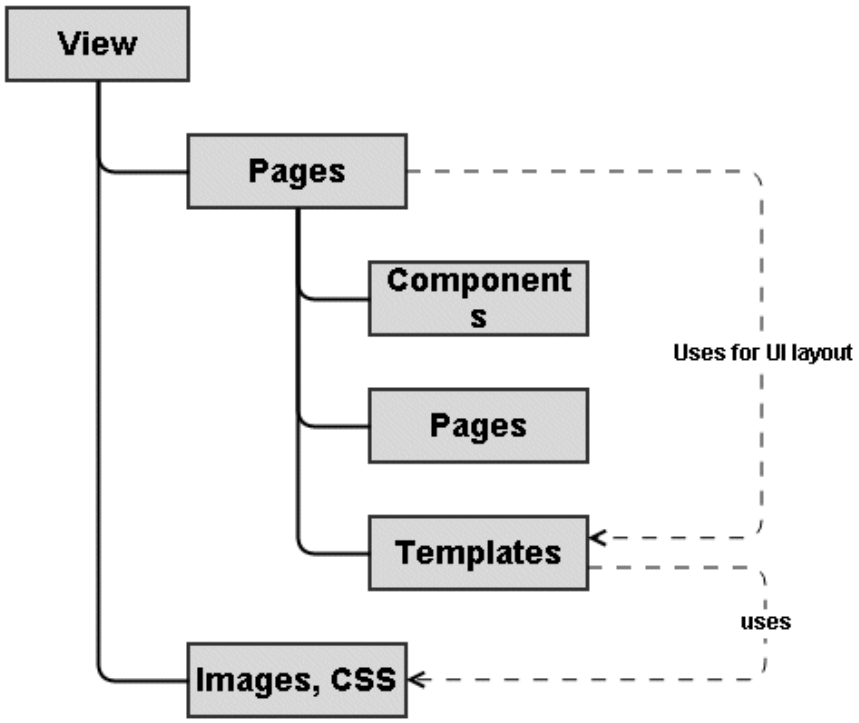
- Navigate to `<API Portal install directory>/API_Portal/server/support` and run the `collectlogfiles.bat` file

  A zip file logfiles.zip with the logs is created in the same folder. The zip file contains various folders where corresponding relevant API Portal log files are present.

16 Customizing API Portal Views

API Portal Views

In the Views page, a list of views (system and user-defined) is displayed. A view (for example, mysagtours) contains a set of pages, templates, components, and a set of other supporting resources. A page is a logical container of all the elements in a user interface page. The page with the help of templates defines the layout of the elements in a user interface. A page in turn can contain inner pages and components. The components are operational elements in the user interface. For example, user interface (UI) widgets, buttons, and so on.



You can create a new view and customize it depending on the required layout. You can also restore a view by uploading a backed up file. The table below lists the operations you can perform in the Views page.

Click...	To...
Create	Create a new view.
Restore	Restore a backed up view.

Note:
For detailed information on customizing various sections in views, see *webMethods API Portal Customization Guide*.

Creating a View

API Portal is highly customizable. The portal can be completely re-branded to your company's look and feel. You can change the predefined rendering templates that affect layout, fonts, styles, images, add your own content blocks and style sheets, include java widgets and so on. You can rearrange the navigation or include additional links.

Prerequisite

- You must have the privileges of a **Portal Administrator**.
- You must have knowledge of HTML, CSS, and java script.

To create a new view

1. Click  in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Click **Create**.
6. Type the name of the new view.
7. Select the template, for example **apiportal**.
8. Click **Apply**.

You created a new view.

Before proceeding to customization, the Portal Administrator has to activate the new view.

Activating a View

You can activate a view for displaying the API Portal data. By default, you have the apiportal and sagtours views.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To activate a view

1. Click  in the right top corner of the API Portal window to display the menu options.

2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.

   All available views are displayed. The current view has **(active)** displayed next to the view.

5. Hover the mouse over the relevant view and click **Activate**.

The selected view is now activated and becomes the active view.

Customizing Pages

You can customize the look and feel of pages of a newly created view. This section gives the outline procedure of customizing pages. For detailed procedure on customizing the API Portal Home Page, API Details page and API Gallery page, see *webMethods API Portal Customization Guide*

Prerequisite

- You must have the privileges of a **Portal Administrator**.
- You must have knowledge of HTML, CSS, and java script.
- You should have already created and activated a new view.

To customize pages

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Hover the mouse over the view to be customized and click **Edit**.
6. Click the **Pages** tab to view the list of pages.
7. Click the required page.
8. Click **Basic configuration** to activate/deactivate basic page elements such as name, search, top level navigation, and so on.

9. Click **Advanced configuration** to activate/deactivate other page elements, and to customize the templates and styles.

10. Click **Save**.

You customized an item. Click **Preview** to display a preview of the customized page.

Restoring a Default Item or Page

You can restore the page or an item to its earlier view from its customized view.

Prerequisite

- You must have the privileges of a **Portal Administrator**.
- You must have knowledge of HTML, CSS, and java script.
- You should have a customized view.

To restore the default item or page

1. Click the menu options displayed in the right top corner of the API Portal window.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Hover the mouse over the view to be restored and click **Edit**.
6. Click the **Pages** tab to view the list of pages.
7. Click the item to be restored to its original view. For example,**Home**.
8. Click **Restore original template icon**.
9. Click **Save**.

You restored the default item.

Backing up a View

You can back up a customized view that can be used later.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To back up a view

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Hover the mouse over the view to be backed up and click **Backup**.

You backed up a view.

Restoring a View

You can restore a customized view that was backed up earlier to apply the same settings.

Prerequisite

- You must have the privileges of a **Portal Administrator**.
- You should have backed up the view.

To restore a view

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Click **Restore**.
6. Select the file that contains the backup of the view to be restored and click **OK**.

You restored a view.

Renaming a View

You can rename a view as required.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To rename a view

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Hover the mouse over the view you want to edit.
6. Click **Rename**.
7. Type the new name.
8. Click **Rename**.

You renamed the view.

Deleting a View

You can delete a view that is not required.

Prerequisite

You must have the privileges of a **Portal Administrator**.

To delete a view

1. Click in the right top corner of the API Portal window to display the menu options.
2. Click **Administration**.
3. Click **Customization**.
4. Click **Views**.
5. Hover the mouse over the view you want to edit.
6. Click **Delete**.
7. Click **OK** on the confirmation prompt.

You deleted a view.

17 Remove User Data from API Portal

Removing User Data

Data protection laws and regulations, such as the General Data Protection Regulation (GDPR) might require specific handling of user data, even after a user profile is removed. Additionally, employees or other clients with user accounts on API Portal may request that any user identifying information such as user name, email addresses, or client IP addresses be removed from API Portal. When a user is deleted in API Portal UMC, the audit events still retain the information about the user which should be deleted or anonymized. To comply with data protection requirements and user requests, in addition to deleting the user account, you have to complete activities to remove or anonymize data.

Anonymizing user data in UMC

As an Administrator after deleting a user or a user group, wait for thirty minutes so that the deleted audit is logged. Run the CLI command y-tenantmgmt to anonymize the user and group data.

1. Open command prompt and navigate to the location *Install_directory*\API_Portal\server\bin\work\apiportalbundle_s\tools\bin.

2. Run the following command for anonymizing user data:

```
y-tenantmgmt.bat -s http://hostname:port
-t tenant_name anonymize -u system -p ******* -n username -type user
```

3. Run the following command for anonymizing group data:

```
y-tenantmgmt.bat -s http://hostname:port
-t tenant_name anonymize -u system -p ******* -n groupname -type group
```

Anonymizing user data in ECP

As an Administrator after deleting a user in UMC, perform the following steps to anonymize user data in ECP.

1. Open API landing page, `http://localhost/#default/home` and login as an Administrator.

2. Change the url to `http://localhost/apidocs`.

3. In the ARIS API page, click **ARIS Collaboration API** link.

4. Click **Persons** and expand `/persons/anonymize` resource.

5. In the **body** parameter, provide the username in `{"userNames": ["<user_name>"]}` as value and click the **Try it out!** button.

   You will receive 202 as the response code for success.

   The username is changed to **Anonymized User1** in all the applicable places.

18 API Portal REST APIs

API Portal REST APIs

API Portal provides the capability to administer various functions of the application using REST APIs.

You can access the REST APIs using the following URL: http://localhost/abs/apirepository/<path>. For example, to access the */apis* path, you must enter http://localhost/abs/apirepository/apis.

You can also get to the APIs through the swagger file in the following location of the folder in which you have installed API Portal:
`Install_Dir`\API_Portal\server\bin\work\work_apiportalbundle_s\base\webapps\abs\docs.

Manage APIs

API Portal provides the capability to retrieve the APIs published in the portal, update the APIs, or delete them. API Portal provides the following REST APIs and the resources to manage APIs:

API Details
Path: /apis **Method**: GET **Description**: Retrieves the list of APIs present in API Portal.
Path: /apis/{apidId} **Method**: GET **Description**: Retrieves the details of a specific API.
Path: /apis/{apidId} **Method**: DELETE **Description**: Purges a specific API. You cannot undo this.
Path: /apis/{apidId} **Method**: PATCH **Description**: Modifies the metadata of a specific API.
Path: /apis/{apiId}/tags **Method**: GET **Description**: Retrieves the tags associated with a specific API.
Path: /apis/{apiId}/tags **Method**: POST

API Details
Description: Adds tags to a specific API.
Path: /v1/apis **Method**: POST **Description**: Updates the definition of APIs.
Path: /v1/apis/{apiId} **Method**: POST **Description**: Updates the definition given through payload for a specific API.

Manage Communities

API Portal provides the capability to retrieve the list of communities and the APIs that are part of the required communities. Also, you can add or delete the APIs from a community. API Portal provides the following REST APIs and the resources to manage communities:

API Details
Path: /communities **Method**: GET **Description**: Retrieves the list of communities present in API Portal.
Path: /communities/{community-id} **Method**: GET **Description**: Retrieves the details of a specific community.
Path: /communities/{community-id}/apis **Method**: GET **Description**: Retrieves the list of APIs associated to a specific community.
Path: /communities/{community-id}/apis **Method**: POST **Description**: Adds APIs to a specific community.
Path: /communities/{community-id}/apis **Method**: DELETE **Description**: Removes API from a specific community.

API Details
Path: /communities/{community-id}/ apis/{api-id} **Method**: DELETE **Description**: Removes a specific API from a specific community.
Path: /communities/{community-id}/users **Method**: GET **Description**: Retrieves the list of users that are part of a specific community.

Manage Providers

API Portal provides the capability to add, modify, or delete API providers, associate APIs with providers, and manage the list of APIs associated with a provider. API Portal provides the following REST APIs and the resources to manage API providers:

API Details
Path: /providers **Method**: GET **Description**: Retrieves the list of API Providers registered in API Portal.
Path: /providers **Method**: POST **Description**: Registers an API provider in API Portal.
Path: providers/status/{handlerId} **Method**: GET **Description**: Retrieves the status of the providers in API Portal.
Path: /providers/{providerId} **Method**: GET **Description**: Retrieves the details of a specific provider.
Path: /providers/{providerId} **Method**: PUT **Description**: Updates the details of a specific provider.
Path: /providers/{providerId}

API Details
Method: DELETE **Description**: Removes a specific provider from API Portal.
Path: /providers/{providerId}/apis **Method**: GET **Description**: Retrieves the API(s) linked to a specific provider.
Path: /providers/{providerId}/apis **Method**: PUT **Description**: Updates the API(s) linked to a specific provider.
Path: /providers/{providerId}/apis **Method**: DELETE **Description**: Unlinks the API(s) linked from a specific provider.
Path: /providers/{providerId}/apis/{apiId} **Method**: PUT **Description**: Links a specific API with a specific provider.
Path: /providers/{providerId}/apis/{apiId} **Method**: DELETE **Description**: Unlinks a specific API from a specific provider.

Manage Applications (Access Tokens)

API Portal provides the capability to retrieve the list of applications, renew or delete the access token required to access an application, view the list of events for APIs, and update the status of events. API Portal provides the following REST APIs and the resources to manage applications and events:

API Details
Path: /v1/accesstokens **Method**: GET **Description**: Retrieves the list of applications for the currently logged-in user.
Path: /v1/accesstokens **Method**: POST

API Details
Description: Creates an application for the given event Id.
Path: /v1/accesstokens/{appId} **Method**: PUT **Description**: Renews the access tokens of a specific application associated with the event Id.
Path: /v1/accesstokens/{appId} **Method**: DELETE **Description**: Deletes the access tokens of a specific application associated with the event Id.
Path: /v1/events **Method**: GET **Description**: Retrieves the list of events in API Portal filtered by query parameters.
Path: /v1/events **Method**: POST **Description**: Updates the status of the list of events in API Portal.
Path: /v1/events/apis **Method**: POST **Description**: Retrieves the list of events for all APIs published in API Portal.

Export API Usage Details

API Portal provides the capability to download the API usage of a specific user through the following API:

API Details
Path: /runtimeevents/export **Method**: GET **Description**: Retrieves the API usage details of the logged-in user, filtered by applications and APIs.