webMethods API-Portal Administrator’s Guide

Version 9.7

October 2014

This document applies to webMethods API-Portal Version 9.7 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-97-20141205

Table of Contents

About this Guide....................5
- Document Conventions.................... 5
- Documentation Installation.................... 6
- Online Information.................... 6

Overview.................... 7
- Why Do Organizations Expose APIs?....................8
- Why Do APIs Need to Be Managed?.................... 8
- What is webMethods API-Portal?....................9

Configuring API-Portal.................... 13
- Before You Configure API-Portal....................14
- Security Considerations.................... 14
  - Securing Client Requests.................... 14
  - Preventing Use of the HTTP OPTIONS Method....................14
  - Implementing Secure Password Policies.................... 15
- Sending Email Notifications....................15
  - Configuring the SMTP Mail Server Connection for API-Portal....................16
- Configuring Ports.................... 17
  - About Ports....................17
  - Default Ports....................17
  - About Enabling/Disabling a Port.................... 17
    - Disabling a Port....................18
    - Enabling a Port....................18
  - Testing for HTTPS Requests....................18
- Considerations for Machines with Multiple Network Interfaces.................... 18
  - Reconfiguring the Load Balancer....................19
- Roles Needed to Perform Configuration Tasks....................20
- Configuring CentraSite, Mediator, and API-Portal....................20
- Preparing to Publish APIs to API-Portal....................22
- API-Portal License File....................23
  - Importing the API-Portal License File using the ACC.................... 24
  - Importing the API-Portal License File using the UMC.................... 24

Managing API-Portal....................27
- Overview.................... 28
- What Happens When You Start API-Portal?....................28
- Starting API-Portal (Windows)....................28
- Starting API-Portal (Linux/UNIX).................... 29
- Stopping API-Portal (Windows).................... 29
- Stopping API-Portal (Linux/UNIX).................... 30
- Opening the API-Portal User Interface in a Browser.................... 30

Starting and Stopping the API-Portal Cloud Controller (ACC).. 31
Verifying the Status of API-Portal Components.. 31
Understanding API-Portal Component Status in ACC.. 32
API-Portal Components... 33
Starting and Stopping API-Portal Components... 34
Gathering Diagnostic Information... 34

Managing Users.. 37
Overview.. 38
Users, Groups, Roles, and Permissions in CentraSite... 38
Predefined Roles in CentraSite..38
Other Users in CentraSite.. 40
Predefined Roles in API-Portal..41
Other Users and Roles in API-Portal.. 41

Managing API Assets... 43
Planning for API Management.. 44
About API-Portal Assets.. 44
API-Portal Profile in CentraSite..44
Publishing and Unpublishing APIs to and from API-Portal... 45
Handling Requests for API Access Tokens...46

API-Portal Configuration Parameters.. 47
Overview.. 48
apiportal.analytics.. 48
apiportal.proxy..48
apiportal.geo.. 49

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, whether inside your organization or outside to partners and third parties. In addition to describing the API management components and workflow, the guide explains how to configure API-Portal for use with CentraSite and webMethods Mediator, 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 storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
`Monospace font`	Identifies text you must type or messages displayed by the system.
{ }	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.

Convention	Description
[ ]	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 (...).

Documentation Installation

You can download the product documentation using the Software AG Installer. The documentation is downloaded to a central directory named _documentation in the main installation directory (SoftwareAG by default).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

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 and certified samples, go to Products.

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

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, 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.

1 Overview

- Why Do Organizations Expose APIs? 8
- Why Do APIs Need to Be Managed? 8
- What is webMethods API-Portal? 9

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 chooses to expose contain core assets the organization would naturally 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 will 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.

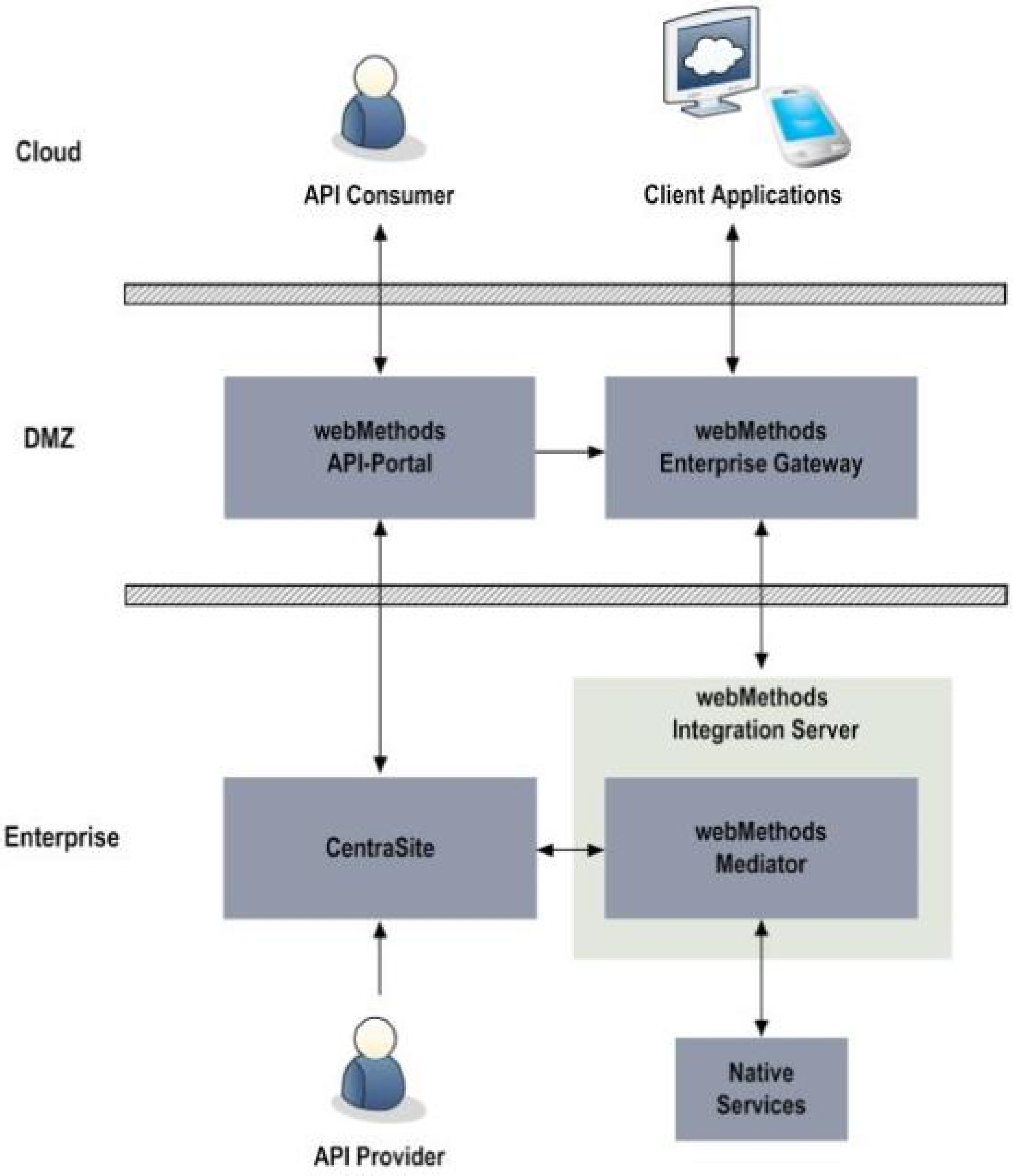
- Collaborate with other API consumers by way of forums or integration with social media.

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 apps 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 adding 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 onboarding of new users**. Approval workflows in CentraSite simplify the onboarding process by provisioning 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.
- **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 information about where visitors are coming from, how many visitors become registered users, how many provisioned access tokens are actually used, what pages gather the most interest, and which APIs are more popular than others. This information is available by way of dashboards to API providers who have an API administrator role in API-Portal. With this information, providers can understand how their APIs are being used, which in turn can help identify ways to improve their 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 invoke a protected 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. For details about installing the SAGTours demo and using it as a basis for customizing your own portal, see *webMethods API-Portal Online Help*.

- **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 keys and OAuth2 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 API-Portal and the 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.

2 Configuring API-Portal

- Before You Configure API-Portal 14
- Security Considerations 14
- Sending Email Notifications 15
- Configuring Ports 17
- Considerations for Machines with Multiple Network Interfaces 18
- Roles Needed to Perform Configuration Tasks 20
- Configuring CentraSite, Mediator, and API-Portal 20
- Preparing to Publish APIs to API-Portal 22
- API-Portal License File 23

Before You Configure API-Portal

Before you start configuring webMethods API-Portal, make sure the following products are installed in addition to API-Portal:

- CentraSite
- webMethods Integration Server
- webMethods Mediator
- webMethods Enterprise Gateway (optional)

If you are using Enterprise Gateway to secure traffic between API consumer requests and the execution of services on Mediator, ensure that CentraSite, Integration Server, and Mediator are installed behind the firewall. In this scenario, the Enterprise Gateway Server is installed in the DMZ for external access. Alternatively, for internal users only, you can use Integration Server with Mediator deployed in a secure network behind the firewall.

For more information about installing these products, see *Installing webMethods and Intelligent Business Operations Products*.

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 18.

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 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_m\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 UMC application at: http://*host* :*port* /umc
2. Click **Configuration**, and then select **Password policy** from the drop-down list to display all the related properties.
3. The current password policy settings are shown. Change the properties as needed. To see a description of each property, hover your cursor over the property name.

For additional information about setting passwords and using the Configuration page, see the *webMethods API-Portal Online Help*.

Sending Email Notifications

Both API-Portal and CentraSite 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. Additionally, during access token requests, if CentraSite is not reachable, API-Portal will send an email alert to all users with the API-Portal Administrator role.

CentraSite can send email notifications about:

- Access tokens expiring
- Policy-related actions
- Service deployment

For CentraSite to issue email messages, an administrator must first configure CentraSite's email server settings. CentraSite also provides predefined email templates for use with API key or OAuth token generation, renewal, and expiration. By default, these templates are configured in the centrasite.xml file. But, if you do not want to use the predefined email templates, you can create your own templates and configure the centrasite.xml file as necessary. For instructions to configure CentraSite email server settings, see the *CentraSite Administrator's Guide*. For more information about the predefined email templates, see the section on managing API keys and OAuth 2.0 tokens in *Working with the CentraSite Business UI*.

Configuring the SMTP Mail Server Connection for API-Portal

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

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
   ```

2. Set the sender's email address:

   a. Start a web browser and go to the API-Portal UMC application:

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

   b. Click **Configuration**, and then select **SMTP** from the drop-down list to display all the related parameters.

c. Double click the com.aris.umc.notification.sender parameter and enter your mail server address. For example:

```
com.aris.umc.notification.sender=API-Portal@MyCompany.com
```

3. In the ACC, restart API-Portal:

   a. Enter `stopall` to stop API-Portal.

   b. Enter `list` to verify the status of the API-Portal components and ensure that all are in the STOPPED state before proceeding.

   c. After all components have stopped, enter `startall` to start API-Portal.

Configuring Ports

About 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.

Default Ports

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

About Enabling/Disabling a 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 Integration Server 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 enter:

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

4. Start the load balancer component.
5. To verify you have deactivated the port, try logging in. The welcome screen will not be 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 enter:

```
reconfigure loadbalancer_m  +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

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, enter the `show instance loadbalancer_m` command to see the current configuration settings of the load balancer component. For example:

```
ACC+ localhost>show instance loadbalancer_m
ID: loadbalancer_m state:STARTED type:com.aris.runnables.httpd.httpd-run-
prod-97.0.2.2)
   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.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:18050
```

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_m
   ```

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

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

   c. Start the load balancer component:

   ```
   start loadbalancer_m
   ```

Roles Needed to Perform Configuration Tasks

To add and manage API-Portal instances in CentraSite, you must belong to the CentraSite Administrator role or at least the API-Portal Administrator role.

- If you belong to the CentraSite Administrator role, you can manage any API-Portal within any organization.
- If you belong to the API-Portal Administrator role for an organization, you can manage all of the API-Portal instances in that particular organization.

  The API-Portal administrator is responsible for installing, configuring, and maintaining API-Portal. The administrator is also responsible for ensuring the server is secure, available to clients, and running at peak performance. Usually, one person is appointed as the API-Portal administrator, although most sites identify at least one other person to act as a backup.

For more information about roles, see "Managing Users" on page 37 and the *CentraSite Administrator's Guide*.

Configuring CentraSite, Mediator, and API-Portal

After the components listed in "Before You Configure API-Portal " on page 14 are installed, additional configuration tasks must be performed to set up the environment in preparation for publishing APIs to API-Portal. The following table lists these high-level configuration steps and where to go for more information:

Step	Where to Go for Procedures
Import the API-Portal license file.	" API-Portal License File" on page 23
Configure the CentraSite infrastructure, including LDAP, email server, and log purging.	*CentraSite Administrator's Guide*
Set up an organization structure for API provider and consumer organizations.	*Working with the CentraSite Business UI*
(Optional) Configure Mediator in a clustered Integration Server environment.	Section on configuring communication with CentraSite in *Administering webMethods Mediator*

Step	Where to Go for Procedures
Configure API-Portal email SMTP server and sender'e email address.	"Configuring the SMTP Mail Server Connection for API-Portal " on page 16
Configure the communication link between Mediator and Enterprise Gateway.	Section on configuring Enterprise Gateway in *webMethods Integration Server Administrator's Guide*
Configure the communication link between CentraSite and Mediator by defining Mediator targets in CentraSite.	*CentraSite Administrator's Guide*
Create a technical user in CentraSite. Specify this user when registering an API-Portal instance in CentraSite. 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 Mediator and API-Portal.	*Working with the CentraSite Business UI*
Create a technical user for Mediator and add the user to the Administration group so that it may be used by CentraSite when publishing APIs to Mediator. **Note:** Software AG recommends specifying the same user credentials as the technical user created in CentraSite and API-Portal.	Section on adding user accounts and adding users to a group in *webMethods Integration Server Administrator's Guide*
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. It is considered a best practice not to tie critical actions (such as publishing	*webMethods API-Portal Online Help*

Step	Where to Go for Procedures
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 and Mediator.	
Create API Provider users (developers who can publish APIs to API-Portal) and assign the users to the API-Portal Administrator role.	Sections on adding a user, assigning a user to a group, and assigning a role to a user in *CentraSite Administrator's Guide*
Register API-Portal instances with CentraSite.	*Working with the CentraSite Business UI*
Configure settings for proxy server, analytics, and geolocation.	" API-Portal Configuration Parameters" on page 47
Set up and customize email templates to be used when informing users about access token request, renewal, and expiration.	Section on managing API keys and OAuth 2.0 tokens in *Working with the CentraSite Business UI*
Create taxonomies in CentraSite that you can use to categorize APIs in the API-Portal Gallery page.	*CentraSite Administrator's Guide*
Customize the API-Portal branding and set up API-Portal templates.	*webMethods API-Portal Online Help*

Preparing to Publish APIs to API-Portal

Before API providers can publish APIs to API-Portal, some additional steps are required. The following table lists these high-level steps and where to go for more information:

Step	Where to Go for Procedures
(Optional) Configure design-time and change-time policies using the predefined policies Publish to API-Portal and UnPublish from API-Portal.	*Working with the CentraSite Business UI*

Step	Where to Go for Procedures
This step is needed only if your organization requires design-time governance.	
Set up approval and onboarding workflows in CentraSite.	Section on working with design/change-time policies in *Working with the CentraSite Business UI*
Create the SOAP or REST API and attach any supporting documents to this asset.	*Working with the CentraSite Business UI*
Create a new virtual alias for the API.	Section on creating a virtual API in *Run-Time Governance with CentraSite*
Configure the run-time policies to enforce for the API you want to expose.	*Run-Time Governance with CentraSite*
Configure consumption settings for the API.	Section on configuring the API consumption settings in access key management policies in *Working with the CentraSite Business UI*
Deploy the API to Mediator.	Section on publishing an API to Mediator in *Run-Time Governance with CentraSite*
Publish the API to API-Portal.	Section on publishing an API to API-Portal in *Run-Time Governance with CentraSite*

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 can use API-Portal. There are two ways to import the license and then get API-Portal up and running: from the API-Portal Cloud Controller (ACC) console or from the API-Portal User Management Console (UMC).

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. At the command line, run the `umcadmin` service, where *license_file* is the full path to your license file:

   ```
   enhance umcadmin_m with importLicense local file license_file options
   tenant.name=default
   ```

5. Enter `stopall` to stop API-Portal.
6. Enter `startall` to restart API-Portal.

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

- "Starting API-Portal (Windows)" on page 28
- "Starting API-Portal (Linux/UNIX)" on page 29
- "Starting and Stopping the API-Portal Cloud Controller (ACC)" on page 31
- "Stopping API-Portal (Windows)" on page 29
- "Stopping API-Portal (Linux/UNIX)" on page 30

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.
5. Stop API-Portal.
6. Start API-Portal.

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

- "Starting API-Portal (Windows)" on page 28
- "Starting API-Portal (Linux/UNIX)" on page 29
- "Stopping API-Portal (Windows)" on page 29

- "Stopping API-Portal (Linux/UNIX)" on page 30

3 Managing API-Portal

- Overview 28
- What Happens When You Start API-Portal? 28
- Starting API-Portal (Windows) 28
- Starting API-Portal (Linux/UNIX) 29
- Stopping API-Portal (Windows) 29
- Stopping API-Portal (Linux/UNIX) 30
- Opening the API-Portal User Interface in a Browser 30
- Starting and Stopping the API-Portal Cloud Controller (ACC) 31
- Verifying the Status of API-Portal Components 31
- Understanding API-Portal Component Status in ACC 32
- API-Portal Components 33
- Starting and Stopping API-Portal Components 34
- Gathering Diagnostic Information 34

Overview

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 and is set to start automatically when the machine on which it is installed initializes. You do not have to manually restart API-Portal following a machine restart.

On Linux systems, you need to manually start API-Portal.

When you start API-Portal for the first time after installation, the following items are created:

- A default tenant (which includes loading the initial tenant database)
- License file imports/assignments
- Default roles for API-Portal Administrator, API Provider, and API Consumer

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 will issue 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, enter the `startall` command.

2. Verify the status of all API-Portal components. For details, see "Verifying the Status of API-Portal Components" on page 31.

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 will issue 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
   - 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 will prompt you for it.

   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
   ```

2. At the ACC command prompt, enter the `startall` command.
3. Verify the status of all API-Portal components. For details, see "Verifying the Status of API-Portal Components" on page 31.

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, enter the `stopall` command.

2. Verify the status of all API-Portal components. For details, see "Verifying the Status of API-Portal Components" on page 31.

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
    - 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 will prompt you for it.

    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
    ```

2. At the ACC command prompt, enter the `stopall` command.
3. Verify the status of all API-Portal components. For details, see "Verifying the Status of API-Portal Components" on page 31.

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, your browser will issue 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. When API-Portal loads in the browser, log in with your user name and password.

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

   - User Name: system
   - Password: manager

Starting and Stopping the API-Portal Cloud Controller (ACC)

The API-Portal Cloud Controller (ACC) is a command line tool that allows starting, stopping, administrating, and advanced customizing of API-Portal components.

To start or stop the ACC

1. If you are using Windows, do the following:

   a. Start the ACC using the following Windows shortcut:

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

   b. In the command window, use the `start` or `startall` commands to start the API-Portal components and the `stop` or `stopall` commands to stop the API-Portal components.

2. If you are using Linux or UNIX, do the following:

   a. Navigate to the following directory:

      Software AG_directory/API_Portal/server/acc/

   b. Start the ACC by issuing the command `acc.sh`.

   c. In the command window, use the `startall` command to start the API-Portal components and the `stopall` command to stop the API-Portal components.

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 there are two components (abs_m and copernicus_m) with a status of STARTING.

```
ACC+ localhost>list
On node localhost 11 runnables are installed.
zoo_m          : STARTED (com.aris.runnables.zookeeper-run-prod-1.0.0-RC22-
Trunk-SNAPSHOT)
postgres_m     : STARTED (com.aris.runnables.PostgreSQL-run-prod-1.0.0-RC22-
Trunk-windows64-SNAPSHOT)
```

```
couchdb_m      : STARTED (com.aris.runnables.couchdb-run-prod-1.1.1-RC22-Trunk-
windows-SNAPSHOT)
cloudsearch_m  : STARTED (com.aris.cip.y-cloudsearch-run-prod-3.0.0-RC32-Trunk-
SNAPSHOT)
elastic_m      : STARTED (com.aris.runnables.elasticsearch-run-prod-1.0.0-RC22-
Trunk-SNAPSHOT)
umcadmin_m     : STARTED (com.aris.umcadmin.y-umcadmin-run-prod-12.0.0-RC29-
Trunk-SNAPSHOT)
adsadmin_m     : STARTED (com.aris.adsadmin.y-adsadmin-run-prod-12.0.0-RC29-
Trunk-SNAPSHOT)
ecp_m          : STARTED (com.aris.runnables.ecp-run-prod-3.0.0-RC27-Trunk-
SNAPSHOT)
abs_m          : STARTING (com.aris.modeling.components.y-server-run-prod-9.0.0-
API2-Dev-SNAPSHOT)
copernicus_m   : STARTING (com.aris.copernicus.copernicus-portal-server-run-
prod-9.0.0-features-API2-SNAPSHOT)
loadbalancer_m : STARTED (com.aris.runnables.httpd.httpd-run-prod-1.0.0-RC22-
Trunk-windows64-SNAPSHOT)
```

For more details about the API-Portal components and their status in the ACC, see " API-Portal Components" on page 33 and "Understanding API-Portal Component Status in ACC" on page 32.

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 will attempt 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.

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_m	Service registry
postgres_m	Primary persistence storage for API-Portal
couchdb_m	Document store persistence
cloudsearch_m	Intelligent API search capabilities for API-Portal
elastic_m	Search storage engine with search capabilities
umcadmin_m	User management server
adsadmin_m	Document store administration server
ecp_m	Enterprise Collaboration Platform
abs_m	API-Portal business logic
copernicus_m	API-Portal user interface
loadbalancer_m	Load balancer to distribute the request load across servers

For more information about the ACC, see the following topics:

- "Starting and Stopping the API-Portal Cloud Controller (ACC)" on page 31
- "Verifying the Status of API-Portal Components" on page 31

Starting and Stopping API-Portal Components

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

Start the ACC and enter 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.
`start` *`instanceId`*	Starts the specified API-Portal component. For example, the command to start the abs_m component is: `start abs_m`
`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 abs_m component is:
	`stop abs_m`
	If successful, the system responds:
	`Successfully stopped runnable abs_m on node localhost`
	If issues arise, ACC returns additional information. For example:
	`Could not stop runnable abs_m on node localhost: Ant stop exited with 1`

Gathering Diagnostic Information

API-Portal maintains a number of logs files that may be helpful to Software AG Global Support when they are analyzing issues.

- When running API-Portal on Windows, run `collectlogfiles.bat` to gather all the log files. You can find the file at this location:

Software AG_directory\API_Portal\server\support\collectlogfiles.bat

- When running API-Portal on Linux, use the Linux `find` command across the *Software AG_directory*\API_Portal\server directory and locate all files that contain *.log.

4 Managing Users

- Overview 38
- Users, Groups, Roles, and Permissions in CentraSite 38
- Predefined Roles in CentraSite 38
- Other Users in CentraSite 40
- Predefined Roles in API-Portal 41
- Other Users and Roles in API-Portal 41

Overview

API-Portal user management is done in CentraSite and in API-Portal User Management Console (UMC). In CentraSite, user management centers more around allowing and controlling access to who can do certain things regarding assets and objects. In API-Portal, the focus is on who can publish the APIs to API-Portal and access API-Portal itself.

Users, Groups, Roles, and Permissions in CentraSite

Users identify individuals that are known to CentraSite. The roles and permissions that are assigned to users specify which operations they can perform and which registry objects they can access.

A *group* describes a set of CentraSite users. The group always belongs to exactly one organization, but it can contain users from different organizations. Groups are visible to all users.

In CentraSite, access control is enabled through a system of permissions and roles. The roles to which you belong and the permissions they include dictate what portions of the CentraSite user interface you see, what objects you can work with, and what operations you can perform.

- A *permission* enables a user to perform a specified operation on a specified object. Permissions also enable users to work with specified parts of the CentraSite user interface.
- A *role* is a collection of permissions that can be assigned to an individual user or a group.

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

Predefined Roles in CentraSite

Roles in CentraSite can be assigned to users or groups defined in an organization. Users or groups who have roles receive all of the permissions associated with the roles. CentraSite provides some predefined roles for you to use. You can also create custom roles as needed.

For complete information about the predefined roles and creating custom roles in CentraSite, see *CentraSite Administrator's Guide*.

Predefined CentraSite Role	Description
CentraSite Administrator (CSA)	System-level role. This role includes all available permissions. Users in this role have all the permissions to manage the complete system (that is, users with “superuser” permissions). Only a CentraSite Administrator can change the permission for a system role. Additionally, the CSA can create and delete a system role. The CentraSite Administrator role cannot be modified or deleted.
Organization Administrator	Organization-level role. Users in this role can manage objects within a particular organization. This includes all child organizations of that organization. The Organization Administrator role cannot be modified or deleted.
Asset Provider	Organization-level role. Users (providers of the API) in this role can create and view assets within his or her organization. This role also has the ability to register users to consume assets. By default, all CentraSite users in an organization belong to this role.
Asset Consumer	Organization-level role. Users (consumers of the API) in this role can only see the set of APIs that providers have given them permission to view. Additionally, they can invoke (call) the APIs exposed to them. By default, all CentraSite users belong to this role, giving them permission to view the assets for their organization.
API Runtime Provider	Organization-level role. Users in this role can manage the API lifecycle, including API creation and documentation, API virtualization (publishing to Mediator), policy

Predefined CentraSite Role	Description
	enforcement, and API analytics. These users can also create and register API-Portal instances in CentraSite.
API Publisher	Organization-level role. Users in this role can publish run-time policies within an organization. This role also has the ability to publish APIs to gateways, such as to Mediator.
API-Portal Administrator	Organization-level role. Users in this role can manage API-Portal instances within an organization. This role is required to publish APIs to API-Portal.

For more information, see the section on getting started with API Management in *Working with the CentraSite Business UI*.

Other Users in CentraSite

The CentraSite user account to which you belong is either an API provider or an API consumer:

- **API providers** (owners of the API) who belong to the API Runtime Provider role in CentraSite are allowed to create and manage (view, edit, delete) all APIs within their organization. Additionally, these users can virtualize and publish APIs to the run-time layer.
- **API consumers** (users of the API) who belong to the Asset Consumer role in CentraSite can only work with the set of APIs that API providers have given them permission to view. Additionally, API consumers can invoke (call) the APIs exposed to them.

If your API management lifecycle includes approval workflows, some CentraSite users can be designated as an approver. Approvers whose user account includes a valid email address can receive an email message informing them that a request is awaiting their approval. Approvers can be in a designated approver group, and manage various aspects of the approval lifecycle, including review and approval or rejection of:

- Onboarding requests for API-Portal
- API access token requests (new, renew, revoke)

In addition to these users, 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). CentraSitean API-PortalCentraSite

Predefined Roles in API-Portal

Roles in API-Portal can be assigned to users or groups defined in an organization. Users or groups who have roles receive all of the permissions associated with the roles.

API-Portal provides some predefined roles for you to use. You can also create custom roles as needed.

The following is a partial list of the roles and function privileges in API-Portal that apply to API users and administration. For a complete list of all function privileges that can be assigned, see *webMethods API-Portal Online Help*. 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.

Predefined API-Portal Role	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.
Guest	A guest user is someone who comes to the API-Portal and is looking for APIs. These anonymous users can browse and test the available APIs. Once a guest user decides to use an API, the user must register and request an access token. **Important:** Do not change the credentials for the Guest user. Username/password credentials must remain guest/guest. In the API-Portal UI, changing the guest user password in a user's properties page of UMC causes API-Portal UI page components to not render correctly or to not render the API-Portal at all.

Other Users and Roles in API-Portal

The API-Portal user account to which you belong is either an API provider or an API consumer:

- **API provider.** Users in this group are allowed to publish APIs to API-Portal. These users are registered in CentraSite and APIs are published to API-Portal.
- **API consumer.** Users in this group are allowed to browse the portal (anonymously or as a registered user), register as an API-Portal user, request API access tokens, and test (evaluate) available APIs.

  After receiving an onboard approval email with an access token, registered developers can build their own applications, including the token in the application.

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.

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

5 Managing API Assets

- Planning for API Management 44
- About API-Portal Assets 44
- API-Portal Profile in CentraSite 44
- Publishing and Unpublishing APIs to and from API-Portal 45
- Handling Requests for API Access Tokens 46

Planning for API Management

API-Portal functions as the key component of an effective API management solution, along with an instance of CentraSite. CentraSite uses 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 will need?
- Which organizations will API-Portal use?
- Which users in the organization will be using API-Portal to consume its published APIs?
- Which taxonomies and categories are needed to organize the APIs?

About API-Portal Assets

API metadata is stored in CentraSite for each registered API-Portal. For each API-Portal, there is an API-Portal object registered. The API-Portal object has the type of API-Portal, which is a virtual type of type Gateway. An API-Portal is associated with an Organization. Multiple API-Portal instances can share the same Organization.

When an API is published to API-Portal, CentraSite creates a relationship between the API and the API-Portal object in the repository. An API can be published to multiple API-Portal instances.

When an API is unpublished (removed) from API-Portal, its metadata is deleted from the repository, and the API is no longer available from API-Portal. The API remains in CentraSite.

When an access token for an API is requested, CentraSite creates an AccessToken object in the repository and deploys it to the Mediator for run-time policy enforcement. This object is associated with the API-Portal object through a requested relationship. For the requesting user, a User object is created within the Organization associated with API-Portal. The User object is created as soon as a user requests an access token for the first time. The User object is deleted when all its AccessTokens are gone.

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 the 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 freely 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.
- **List of access tokens.** Specifies the access tokens generated for clients who request them through API-Portal.

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 Administrator's Guide*.

The Service asset types (Service, REST Service, and XML Service) and its variants (Virtual Service, Virtual REST Service, and Virtual XML Service) do not contain the **API-Portal Information** profile by default. To enable this profile, an administrator can edit the asset type details and select the **API-Portal Information** profile. For example, an administrator may want to enable this profile for all APIs of asset type Service. For complete instructions about extending asset type definitions to enable the API-Portal profile, see *Working with the CentraSite Business UI*.

Publishing and Unpublishing APIs to and from API-Portal

To publish APIs from CentraSite to API-Portal, you must belong to the CentraSite Administrator role or the API-Portal Administrator role.

- If your user account belongs to the CentraSite Administrator role, you can publish APIs that belong to any organization.
- If your user account belongs to the API-Portal Administrator role for an organization, you can publish APIs that belong to that organization.

For more information about roles and permissions, see *CentraSite Administrator'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 CentraSite repository, and then publishes those bundles to API-Portal.

- **Unpublish from API-Portal.** This action removes the specified API metadata bundle from the API-Portal.

For more information about configuring the design/change-time policies that API-Portal uses, see *Working with the CentraSite Business UI*. For procedures for publishing and unpublishing APIs to and from API-Portal, see *Run-Time Governance with CentraSite*.

Handling Requests for API Access Tokens

When an API developer signs up and requests an API access token, API-Portal sends the API developer’s sign-up/key request to CentraSite.

CentraSite generates a key, sends the key by way of an email to the requesting API developer, and deploys the key to Mediator.

After 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 will call 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 later analysis and dashboarding.

A API-Portal Configuration Parameters

- Overview 48
- apiportal.analytics. 48
- apiportal.proxy. 48
- apiportal.geo. 49

Overview

This appendix contains a description of the parameters you can specify in the API-Portal configuration file (apiportal.properties). This configuration file contains parameters specific to API-Portal data analysis. The file is located in the *Software AG_directory*\API_Portal\server\bin\work\work_abs_m\base\webapps\abs\WEB-INF\classes\com\aris\modeling\server\webapp\apiportal\configuration directory.

To set parameters in the apiportal.properties file, stop API-Portal, and then edit the file directly with a text editor. After you save the changes, restart (start) API-Portal.

The server uses default values for many of the parameters. If a parameter has a default, it is listed with the description of the parameter. If the service request must be routed through a proxy server, these properties specify the proxy server alias for the proxy server through which API-Portal routes the HTTP/S request.

apiportal.analytics.

apiportal.analytics.enabled

Specifies whether API-Portal collects analytics data. When the apiportal.analytics.enabled parameter is set to `true`, API-Portal collects analytics data. When the parameter is set to `false`, API-Portal does not collect analytics data. The default is `true`.

apiportal.analytics.enabled.category

When API-Portal is set to collect analytics data, use the apiportal.analytics.enabled.category property to identify which categories of data will be collected. API-Portal analytics collects three different categories of data: PageViews, UserAudit, and AccessTokenAudit. If you specify multiple categories, separate the names with commas (,).

The default is `PageViews,UserAudit,AccessTokenAudit`

apiportal.proxy.

apiportal.proxy.url

Specifies the proxy server URL used to communicate with the services through the Internet.

Use the format http://*host* :*port*

where *host* is the host name or IP address of the proxy server and *port* is the port number of the proxy server.

apiportal.proxy.username

Optional. If your proxy server requires authentication, use this parameter to specify the username the proxy server uses to authenticate incoming requests.

apiportal.proxy.password

Optional. If your proxy server requires authentication, use this parameter to specify the password the proxy server uses to authenticate incoming requests.

apiportal.geo.

apiportal.geo.service.url

When resolving the location of the user who is accessing API-Portal based on the IP address of the client, API-Portal uses the external service, telize.com, to resolve the location based on the IP address of the client.

The value for this property must be: `http://www.telize.com/geoip/`

If you are using a proxy server, ensure that you also set the properties apiportal.proxy.url, apiportal.proxy.username, and apiportal.proxy.password to ensure API-Portal can communicate the external service.