CentraSite

Virtualizing APIs Using the CentraSite Business UI

Version 9.6

April 2014

This document applies to CentraSite Version 9.6.

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

Document ID: IINM-BU-VIRTUALAPI-96-20140318

Table of Contents

Preface

This document describes how to:

- Create and configure proxy APIs in the CentraSite catalog.
- Publish APIs to the webMethods Mediator.
- Consume APIs using the API key or access token.

API Providers (owners of the API) and API Consumers (consumers of the API) use the CentraSite Business UI to browse and search for APIs in the CentraSite by name, description, attribute values, asset types and/or taxonomy groups. Additionally, the API providers can virtualize and publish APIs to the run-time layer; while the API consumers can invoke (call) the APIs that are exposed to them.

Instructions throughout the remainder of this document use the term "API" when referring to the three base types (Service, XML Service, REST Service), and "API Proxy" when referring to the three virtual types (Virtual Service, Virtual XML Service, and Virtual REST Service) in general.

The content is organized under the following sections:

Browsing and Searching for APIs Describes how to find APIs in the Business UI.

Creating an API Proxy Describes how to create a proxy API.

Viewing Details for an API Proxy Describes how to view the information for proxy APIs in the Business UI.

Editing the properties of an API Proxy Describes how to change a proxy API's information.

Setting Permissions on an API Proxy Describes how to assign instance-level permissions on a proxy API.

Publishing an API Proxy Describes how to publish proxy APIs to webMethods Mediator.

Managing Consumer Applications Describes how to create and manage consumer applications for a proxy API.

Registering as Consumers of an API Describes how users, user groups and consumer applications can register as consumers of a proxy API.

Obtaining Your API Keys and Access Tokens for Consumption Describes how to get your API keys and OAuth access tokens for consuming an API.

Managing Your API Keys Describes how to manage your keys for an API.

Displaying Runtime Information for an API Proxy Describes how to view the runtime information for a proxy API.

Unpublishing an API Proxy Describes how to unpublish a proxy API from webMethods Mediator.

Deleting an API Proxy	Describes how to remove a proxy API through the Business UI.
Privileged User for an API Proxy	Describes information about the privileged user of a proxy API.

1 Browsing and Searching for APIs

The CentraSite Business UI provides a powerful search facility. You can search for APIs across organizations, classifications and types on the basis of several search criteria using a logical conjunction (ALL) or disjunction (ANY) operation. If you need to find APIs based on attributes, then the sophisticated “advanced search” interface also allows you to search against the attributes defined in APIs. The CentraSite Business UI also makes custom fields available as search criteria, allowing you to refine your searches even further.

All CentraSite users, including guests, have permission to browse the CentraSite catalog. You do not need any explicit permission to use the CentraSite catalog in the Business user interface.

Browsing for APIs

You can use the CentraSite’s *Browse* feature to find APIs.

▶ To perform a quick find by browse

1 In CentraSite Business UI, locate the search panel in the top left corner of the navigation bar.

2 From a list of the scopes shown in the drop-down, choose **Assets** to include in the view. The list of scopes includes a set of predefined scopes and custom (i.e. customer-defined) scopes.

3 Click the **Browse** link.

CentraSite returns the APIs in the **Search Results** panel.

4 You can further refine your browse by specifying attributes that are applicable to an API.

For more information about how to use the browse feature for APIs in the Business UI, see the online documentation section *Managing the CentraSite Catalog* > *Browsing the CentraSite Catalog*.

Searching for APIs

You can use the CentraSite's *Search* feature to find APIs.

▶ To search by keyword and scope

1 In CentraSite Business UI, locate the **Type a Search** text box at the top of the screen.

2 From a list of the scopes shown in the drop-down, choose **Assets** to include in the view. The list of scopes includes a set of predefined scopes and custom (i.e. customer-defined) scopes.

3 In the text box, type the keyword(s) to search for. You can use one or more wildcards to specify the keywords.

1. CentraSite applies the filter to the **Name** of the API and returns the first 5 related APIs.

2. Press the Up Arrow and Down Arrow keys to scroll through one API at a time.
3. Locate the API whose details you want to view.
4. Double click or press Enter.
5. If you would like to see more APIs for the specified key pattern, click the **Click here to see more results** link.

4

Else, click the **Search** 🔍 icon.

CentraSite returns the APIs that match both the keyword and scope in the **Search Results** panel.

5 You can further refine your search by specifying attributes that are applicable an the API.

6 You can save your search as a filter in the CentraSite Business UI, so you can recall the same search and run it again or even share it with other users.

For more information about how to use the search feature for APIs in the Business UI, see the online documentation section *Managing the CentraSite Catalog* > *Searching the CentraSite Catalog*.

2 Creating an API Proxy

CentraSite Business UI provides a wizard to create and publish a virtual copy (proxy) of the API.

To create and publish a proxy API, the following prerequisites must be met:

- You must be a registered user in the CentraSite.
- You must belong to one of the following roles:

This Role...	Allows you to...
CentraSite Administrator	Create a proxy API within any organization.
Runtime API Provider	Create a proxy API within a specified organization.
API Publisher	Publish a proxy API into one or more targets.

  For more information about roles and permissions, see the online documentation section *Users, Groups, Roles and Permissions > About Roles and Permissions*.

- You can virtualize an API of any one of the following types in Business UI: Service, XML Service, and REST Service.
- You can publish an API of any one of the following types in Business UI: Virtual Service, Virtual XML Service, and Virtual REST Service.
- Make sure that an API of the type "Service" has a WSDL file, and APIs of the type "XML Service" and "REST Service" have schema files associated to it.
- Ensure that the webMethods Mediator is configured and running on the webMethods Integration Server.

⚠ **Important:** Ensure that the webMethods Mediator property watt.server.auth.skipForMediator Property is set to true. For details, see *Run-Time Governance Reference > Built-In Run-Time Actions Reference for APIs*.

To create and publish a proxy API, you perform the following high-level steps:

Defining a New API Proxy

You use the panel 1 in the **Virtualize** wizard to define a new proxy and endpoint for the API.

To create a new proxy API, the following prerequisites must be met:

- You must belong to either one of the following roles:
  - CentraSite Administrator
  - Runtime Provider
- You must have at least the instance-level Modify permission on that API.

To define a new proxy API

1. Display the details page of the API for that you want to create a virtual copy. For procedures, see the CentraSite online documentation section *API Management Solutions > Viewing Details for an API*.
2. On the API details page, click **Virtualize** to open the **Virtualize** wizard.
3. In panel 1 of the **Virtualize** wizard, specify the following fields:

In this field...	Do the following...
`Create a new virtual alias`	Enter a name for the API proxy. By default, CentraSite populates the proxy API's name with the display name that was specified for native API. If a proxy API with the same name already exists, CentraSite issues a warning message.
`Reconfigure an existing virtual alias`	Alternatively, if a native API has proxy APIs, use this option to adapt it to your requirements.
`Endpoint to be virtualized`	You will see a list of the available endpoints for the native API. Select an endpoint which you want to create as the proxy API by choosing its radio button.

4. Click **Next**.

> **Note:** If you do not enter information into a required field, the system will alert you with a red error icon. Pointing to the icon shows a hint with an error description.

Configuring the Run-Time Actions for an API Proxy

You use the panel 2 of the **Virtualize** wizard to configure the policy governance behavior for a proxy API.

The policy governance behavior of a proxy API is defined by the three components: Policy Accordions, Run-Time Actions, and the Message Flow Stages.

- Summary of the Policy Accordions
- Summary of the Run-Time Actions
- Summary of the Message Flow Stages

Summary of the Policy Accordions

The policy actions are classified into one of the following accordions:

Accordion	Use to...
Request Handling	Request handler allows receiving and transforming the incoming message from a client into a custom format as expected by the native API. For example, a Require HTTP / HTTPS action mandates that the incoming requests for an API are received over the HTTP and/or HTTPS protocol.
Policy Enforcement	Enforces the adherence to real-time policy compliance identifying/authenticating, monitoring, auditing, and measuring and collecting result statistics for an API.
Response Processing	Response handler allows processing of the response message coming from the native API into a custom format as expected by the client.
Error Handling	Error handlers are used to format and return error messages. Errors can occur at run-time for various reasons. For example, security errors occur if a username is not correctly validated or authorized; transformation errors occur if transformation action is unable to successfully transform a message; a routing error is raised if a routing endpoint is unavailable, and so on.

Summary of the Run-Time Actions

Actions are the core elements of stages in a message flow that define the handling of messages as they flow through a proxy API. The following table lists the actions you can configure for the proxy API's message flow, and links you to topics that describe the actions, including how to configure them.

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Request Handling > Protocol			
Require JMS	Specify the JMS protocol for the API to accept and process the requests.	`Receive`	■ SOAP APIs.
Require HTTP / HTTPS	Specify the HTTP and/or HTTPS protocol, SOAP format (for a SOAP-based API), and the HTTP methods (for a REST-based API) for the API to accept and process the requests.	`Receive`	■ SOAP APIs. ■ REST APIs.
Request Transformation	Invoke an XSL transformation in the incoming request before it is submitted to the an API.	`Receive`	■ SOAP APIs. ■ REST APIs.

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Invoke webMethods Integration Server	Invoke a webMethods Integration Server service to pre-process the request before it is submitted to the an API.	`Receive`	■ SOAP APIs. ■ REST APIs.
Policy Enforcement > Authentication			
HTTP Basic Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using HTTP basic authentication mechanism.	`Routing`	■ SOAP APIs. ■ REST APIs.
NTLM Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using NTLM authentication mechanism.	`Routing`	■ SOAP APIs. ■ REST APIs.
OAuth2 Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using OAuth 2.0 authentication mechanism.	`Routing`	■ SOAP APIs. ■ REST APIs.
Policy Enforcement > JMS Routing			
JMS Routing Rule	Specify a JMS queue to which the Mediator is to submit the request, and the destination to which the an API is to return the response.	`Routing`	■ SOAP APIs.
Set Message Properties	Specify JMS message properties to authenticate client requests before submitting to the an APIs.	`Routing`	■ SOAP APIs.
Set JMS Headers	Specify JMS headers to authenticate client requests before submitting to the an APIs.	`Routing`	■ SOAP APIs.
Policy Enforcement > Logging and Monitoring			
Log Invocation	Log request/response payloads to a destination you specify.	`Enforce`	■ SOAP APIs. ■ REST APIs.
Monitor Service Performance	Monitor the run-time performance for a specific consumer, and defines the level of performance that the specified consumer should expect from the API.	`Enforce`	■ SOAP APIs. ■ REST APIs.
Monitor Service Level Agreement	Monitor a user-specified set of run-time performance conditions for an API, and sends alerts to a specified destination when these performance conditions are violated.	`Enforce`	■ SOAP APIs. ■ REST APIs.
Policy Enforcement > Routing			

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Straight Through Routing	Route requests directly to a native endpoint that you specify.	`Routing`	■ SOAP APIs. ■ REST APIs.
Content Based Routing	Route requests to different endpoints based on specific values that appear in the request message.	`Routing`	■ SOAP APIs. ■ REST APIs.
Load Balancing and Failover Routing	Routes the requests across multiple endpoints.	`Routing`	■ SOAP APIs. ■ REST APIs.
Context Based Routing	Route requests to different endpoints based on specific values that appear in the request message.	`Routing`	■ SOAP APIs. ■ REST APIs.
Set Custom Headers	Specify the HTTP headers to process the requests.	`Routing`	■ SOAP APIs. ■ REST APIs.
Policy Enforcement > Security			
Require SSL	Mandate that requests be sent via SSL client certificates, and can be used by both SOAP and REST APIs.	`Enforce`	■ SOAP APIs.
Require Signing	Mandate that a request's XML element (which is represented by an XPath expression) be signed.	`Enforce`	■ SOAP APIs.
Require Encryption	Mandate that a request's XML element (which is represented by an XPath expression) be encrypted.	`Enforce`	■ SOAP APIs.
Require Timestamps	Mandate that timestamps be included in the request header.	`Enforce`	■ SOAP APIs.
Require WSS SAML Token	Uses a WSS Security Assertion Markup Language (SAML) assertion token to validate API consumers.	`Enforce`	■ SOAP APIs.
Evaluate HTTP Basic Authentication	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's authentication credentials contained in the request's Authorization header against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs. ■ REST APIs.

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Evaluate Hostname	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's IP address against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs. ■ REST APIs.
Evaluate IP Address	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's IP address against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs. ■ REST APIs.
Evaluate WSS Username Token	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's WSS username token against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs.
Evaluate WSS X.509 Certificate	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's WSS X.509 token against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs.
Evaluate XPath Expression	■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's XPath expression against the list of users in the Integration Server on which Mediator is running.	`Enforce`	■ SOAP APIs. ■ REST APIs.

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Evaluate OAuth2 Token	- Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). - Validate the client's IP address against the list of users in the Integration Server on which Mediator is running.	`Enforce`	- SOAP APIs. - REST APIs.
Evaluate Client Certificate for SSL Connectivity	- Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). - Validate the client's certificate against the list of users in the Integration Server on which Mediator is running.	`Enforce`	- SOAP APIs. - REST APIs.
Policy Enforcement > Traffic Management			
Throttling Traffic Optimization	- Limit the number of API invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated. - Avoid overloading the back-end services and their infrastructure, to limit specific consumers in terms of resource usage, etc.	`Enforce`	- SOAP APIs. - REST APIs.
Policy Enforcement > Validation			
Validate Schema	Validate all XML request and/or response messages against an XML schema referenced in the WSDL.	`Enforce`	- SOAP APIs.
Response Processing			
Response Transformation	Invoke an XSL transformation in the SOAP response payloads from XML format to the format required by the consumer.	`Response`	- SOAP APIs. - REST APIs.
Invoke webMethods Integration Server	Invoke a webMethods Integration Server service to process the response from the an API before it is returned to the consumer.	`Response`	- SOAP APIs. - REST APIs.
Error Handling			

This action...	Use to...	Available in Message Flow Stage...	Applicable for...
Custom SOAP Response Message	Return a custom error message (and/or the native provider's service fault content) to the consumer when the native provider returns a service fault.	`Response`	■ SOAP APIs.

For more information about the individual run-time actions that are supported out-of-the-box, see the online documentation section *Run-Time Governance Reference* > *Run-Time Actions Reference for APIs*.

Summary of the Message Flow Stages

Message Flow defines the implementation of a (proxy) API.

A message flow is a sequence of stages representing a non-branching one-way processing path. Message flow is used for request, enforce, routing and response paths as well as for error handlers.

A message flow tree is constructed by chaining together actions of these top-level stages:

Stage	Description
Receive	Request stage is used for processing the request path of the Message Flow.
Enforce	Enforce stage is used for processing the enforcement path of the Message Flow. The enforce stage is used to identify and validate specific consumers invoking APIs, throttle traffic, log request/response payloads, and monitor run-time performance conditions.
Routing	Routing stage is used for processing the routing path of the Message Flow. The routing stage is used to perform request-response communication. It represents the boundary between request and response processing for the API. When the routing stage dispatches a request message, request processing is considered finished. When the routing stage receives a response message, response processing begins.
Response	Response stage is used for processing the response path of the Message Flow. In addition, response stage is used as error handler.

To configure the run-time actions for a proxy API, the following prerequisites must be met:

- You must belong to either one of the following roles:
  - CentraSite Administrator
  - Runtime Provider
- You must have at least the instance-level Modify permission on the proxy API.

When you configure the actions for a proxy API, keep the following points in mind:

- When you drag an action from the **Policy Actions** area, the respective step in the **Message Flow** area highlights where the action fits in, this making the navigation from **Policy Actions** area to the **Message Flow** area more intuitive.

- Not all stages support the full set of actions. Every action happens only within a respective step. For example, the "Evaluate" actions occur only on the **Enforce** step; while the"Routing" actions occur only on the **Routing** step.
- Mediator executes the run-time actions configured for a proxy API in a predefined order.
- Some actions are mutually dependent. That is, a specific action must be used in conjunction with another particular action. For example, a **Message Flow** area that includes the Set JMS Headers action must also include the JMS Routing Rule action.
- Some actions are mutually exclusive. That is, a specific action cannot be used in conjunction with another particular action. For example, a **Message Flow** area that includes the JMS Routing Rule action cannot include the Straight Through Routing action.
- Some of the actions are allowed to appear multiple times within a message flow step.

  For those actions that can appear in a message flow only once (for example, Evaluate IP Address), Mediator will choose only one, which might cause problems or unintended results.
- For information about the individual run-time actions that are supported out-of-the-box, see the online documentation section *Run-Time Governance Reference* > *Run-Time Actions Reference for APIs*.
- For information about how Mediator executes actions (and how to avoid policy conflicts), see the online documentation section *Run-Time Governance Reference* > *Built-In Run-Time Actions Reference for APIs* > *Action Evaluation Order and Dependencies* .
- You can view a tooltip text for any accordion by moving the cursor over the accordion name. The tooltip text gives a summary of the accordion’s purpose.
- If you modify the run-time action for a proxy API that is already deployed to webMethods Mediator, CentraSite automatically redeploys the modified API.

Configuring the Run-Time Actions in Message Flow

To configure the run-time actions for a proxy API

1. In the Virtualize page, inspect the accordions in the **Policy Actions** area.
2. Expand an accordion node whose action you want to add to the **Message Flow** area.
3. Drag and drop the required action into the **Message Flow** area.
4. Repeat steps 2 and 3 until you obtain a list of actions in the **Message Flow** area.
5. In the **Message Flow** area, locate the action whose parameter you want to set or examine.
6. Mouse hover the action whose parameters you want to configure.
7. Choose the **Configure** ⚙ icon to the right of the action name.
8. In the <action name> dialog box, set the values for the parameters as necessary.

> **Note:** If you do not enter information into a required field, the system will alert you with a red error icon. Pointing to the icon shows a hint with an error description.

9. Click **OK** to save the parameter settings.

10. You can remove an action from the **Message Flow** area using the **Delete** 🗑 icon to the right of the action name.

11. Click **Next**.

    Or:

    You can choose to perform one of the following actions:

    - **Previous** button to switch back to the previous Create a New Virtual Alias panel.
    - **Virtualize** button to create the proxy API.
    - **Next** button to go to the next Publish the API Proxy panel.
    - **Cancel** button to abandon your unsaved settings.

Publishing an API Proxy

You use the panel 3 of the **Virtualize** wizard to define the targets and publish the proxy API to webMethods Mediator.

To publish a proxy API, the following prerequisites must be met:

- You must belong to either one of the following roles:
  - CentraSite Administrator
  - API Publisher
- You must have at least the instance-level Full permission on the proxy API.
- webMethods Mediator must be configured and running on the webMethods Integration Server.
- To expose a proxy API for consumption, make sure that the " Set API Publish Permissions" policy is active. If the policy is in the inactive state, you must activate it. By default, this policy is active.

  The "Set API Publish Permissions" policy includes a "Set Permissions" action that grants the instance-level “View” permission for the proxy API to the users in “Everyone” group. In addition, you can use the option "Propagate permissions to dependent objects". For more information, see the description of the action in the online documentation section *Built-In Design/Change-Time Actions Reference > Built-In Actions for Design/Change-Time Policies*.

To publish a proxy API

1 From the **Available Targets** list, choose one or more targets in which you want to publish the API.

Or:

If you want to publish the proxy API across all the targets, select the **All Targets** checkbox.

2 In the **Advanced Settings** node, select the **Expose to Consumers** checkbox. This allows unauthorized consumers (guests) to search and access the API.

3 Click **Publish**.

When attempting to publish a proxy API, without selecting at least on target, a warning popup box appears.

Or:

You can choose to perform one of the following actions:

- **Previous** button to switch back to the previous Configure the Run-Time Actions panel.
- **Virtualize** button to create the proxy API, and publish at a later stage. For the description of how to publish a proxy API at a later stage, see the section ***Publishing an API Proxy***
- **Cancel** button to abandon your unsaved settings.

3 Viewing Details for an API Proxy

CentraSite provides a summary of details of the proxy PI. The details rendered as attributes are grouped together as profiles.

To view the details of a proxy API, the following prerequisites must be met:

- The set of proxy APIs that are available to you are the APIs on which you have "View" permission. You can obtain "View" permission on a proxy API in the following ways:
  - By belonging to a role that includes any of the following permissions.

This permission...	Allows you to...
View Assets	View all proxy APIs within a specified organization.
Modify Assets	View and edit all proxy APIs within a specified organization.
Manage Assets	View, edit and delete all proxy APIs within a specified organization, and set instance-level permissions on those APIs. This permission also allows you to create proxy APIs.
Create Assets	Add new proxy APIs to a specified organization. You automatically receive Full permission (which implies Modify and View permission) on all proxy APIs that you create.

  - By having View, Modify or Full instance-level permissions on a proxy API.
- By default, all CentraSite users belong to the “Asset Consumer” role. This role includes the “View Assets” permission for the organization to which a user belongs.

  Having the “Asset Consumer” role gives you implicit view permission on all the proxy APIs in your organization. You can view proxy APIs from other organizations only if you are given permission to do so through the assignment of additional role-based or instance-level permissions.
- In rare instances, an administrator might not grant view permissions to all of the users in an organization. If the administrator of your organization has done this, you will need instance-level permissions on a proxy API in order to view it.

- When you view the details for the API, you will only see profiles for which you have “View” permission.

For more information about permissions, see the CentraSite online documentation section *Users, Groups, Roles and Permissions > About Roles and Permissions* .

Summary of Profiles in the API Proxy

Profile	Description
Basic Information	Contains basic information for a proxy API. This profile shows individual characteristic such as the proxy API's version, type, owner, organization and description. If you are having users watching or consuming the proxy API, CentraSite displays that information on this profile. This profile also contains controls for approving requests placed on this proxy API.
Advanced Information	Contains additional information for a proxy API. This profile shows the technical specifications, and the list of providers and consumers for the proxy API.
Technical Details	Contains technical information for a proxy API. - For a SOAP-based proxy API, it includes the WSDL URL and a list of the operations and bindings. - For an XML/REST-based proxy API, it includes the schema URL and a list of the resources.
Provider Overview	Contains a link to the native API, and a list of Access URIs and API keys.
Consumer Overview	Contains usage information for a proxy API. - For a SOAP-based proxy API, it displays the Consumer Service WSDL / WSDL URL, and a list of Access URIs and API keys. - For an XML/REST-based proxy API, it displays a list of access URIs and API keys.
Identification	*Applicable only for Application and API Key types.* Provides consumer information for a proxy API, including the API key, expiration date of the API key, and the list of consumer identifier tokens associated with the particular API. This profile also contains control for re-configuring the consumer identifier tokens. For a list of the supported consumer identifier tokens, refer to the section ***Using Consumer Identifiers***. **Note:** An Identification profile is displayed for the proxy API ONLY if enabled in the API Key asset type.
API Key Scope	*Applicable only for API Key types.* The name of the a proxy API that is associated with the API key. **Note:** An API Key Scope profile is displayed for the proxy API ONLY if enabled in the API Key asset type.

Profile	Description
Runtime Metrics	*Applicable only for XML/REST-based proxy APIs.* Displays the run-time performance metrics associated with the proxy API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log performance metrics for a proxy API, CentraSite displays those metrics on this profile. **Note:** The Runtime Metrics profile is displayed for proxy APIs regardless of whether you enable or disable in the Service asset type. In other words, when you disable the **Runtime Metrics** profile for the Service asset type, CentraSite removes the profile from the native APIs, but continues to display it for proxy APIs.
Runtime Events	*Applicable only for Virtual XML & REST APIs.* Displays the run-time events associated with the API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log run-time events for an API, CentraSite displays those events on this profile. **Note:** The Runtime Events profile is displayed for proxy APIs regardless of whether you enable or disable in the Service asset type. In other words, when you disable the **Runtime Events** profile for the Service asset type, CentraSite removes the profile from the native APIs, but continues to display it for proxy APIs.

Summary of Actions in the API Proxy

The following actions are available in the details page of a proxy API:

Action Name	Icon	Usage
Save	💾	You use this action to save the change made to an API's information.
Edit	✒	You use this action to view an API's information and modify them.
Version	📄	You use this action to generate a new version of an API.
Delete	🗑	You use this action to remove an API.
Virtualize	🖥	You use this action to create a virtual copy of an API.
Attach	📎	You use this action to attach a supporting document to an API.
Watch	🔭	You use this action to subscribe notifications when changes are made to an API's information.
Add to List	➕	You use this action to add an API to the My Favorites list.
Remove from List	✖	You use this action to remove an API from the My Favorites list.

Action Name	Icon	Usage
Export		You use this action to export an API from one instance of CentraSite to another.
Consume		You use this action to register as a consumer for an API.
Revert		You use this action to revert an API request than has been submitted for approval.
View Report		You use this action to generate and view reports for an API.
Permission		You use this action to assign instance-level and profile-level permissions on an API.
API Consumption Settings		You use this action to configure the client consumption settings for an API.
Publish		You use this action to publish an API to the Mediator for consumption.
Unpublish		You use this action to unpublish an API from the Mediator.

Viewing the Details for an API Proxy

Use the following procedure to view the details for a proxy API.

To view the details of an API

1 In CentraSite Business UI, use either the *Browse* or *Search* feature to locate an API that you want to view. If you need information on how to browse or search the CentraSite catalog, refer to the section *Browsing the CentraSite Catalog* or *Searching the CentraSite Catalog* in the document *Managing the CentraSite Catalog*.

2 Click the API's hyperlinked name.

3 On the APIs' details page, examine the properties as necessary.

CentraSite will display the attributes for the selected API. If you have Modify permission on the API, you can edit the API's attributes.

4 Editing the Properties of an API Proxy

This section describes how to view the information stored for a proxy API and how to change them.

To edit a proxy API, the following prerequisites must be met:

- If you are not the owner of the API, you cannot edit the proxy API unless you have "Modify" permission on that API (granted though either a role-based permission or an instance-level permission).
- When you view the details for the API, you will only see profiles for which you have "View" permission. You will only be able to edit the profiles on which you have "Modify" permission.
- Some attributes are designed to be read-only and cannot be edited even if they appear in a proxy API on which you have "Modify" permission.
- Some attributes accept only specific types of information.

▶ To edit the information for a proxy API

1 In CentraSite Business UI, display the detail page of the proxy API whose attributes you want to modify. If you need procedures for this step, see ***Viewing Details for an API Proxy***.

2 To modify the basic attributes, such as, Name, Description or Version number, place the cursor in the appropriate field and modify the text as required.

3 To modify the extended attributes associated with the proxy API, do the following:

   1. Select the profile that contains the attribute(s) that you want to modify.
   2. Edit the attributes on the profile as necessary.
   3. Repeat steps 3.a and 3.b for each profile that you want to edit.

4 To modify the run-time actions configured for the proxy API, do the following:

   1. Go to panel 2 of the **Virtualize** wizard.
   2. To add an action to the **Message Flow** area, do the following:
      a. In the **Policy Actions** area, expand the desired accordion (Request Handling, Policy Enforcement, Response Handling or Error Handling).
      b. Locate the action you want to add for the proxy API.
      c. Drag and drop the action in the appropriate stage (Receive, Enforce, Routing or Response) in **Message Flow** area.
      d. Repeat the above steps for each action that you want to add.
   3. To reconfigure an existing action in the **Message Flow** area, do the following:
      a. In the **Message Flow** area, locate and mouse hover the action whose parameters you want to modify.

b. Choose the **Configure** ⚙ icon to the right of the action name.

c. In the <action name> dialog box, set the values for the parameters as necessary.

d. Click **OK** to save the parameter settings.

e. Repeat the above steps for each action that you want to modify.

5 Remove an action from the **Message Flow** area using the **Delete** 🗑 icon to the right of the action name.

6 To publish or unpublish the API proxy on or more targets, do the following:

1. Go to panel 3 of the **Virtualize** wizard.
2. Select or unselect the targets from that you want to publish or unpublish the API proxy.
3. Click **Publish** or **Unpublish** button, appropriately.

Profile Information for an API Proxy

You have the following profile information specific to a proxy API:

- Provider Overview
- Consumer Overview Profile
- Identification Profile
- API Key Scope Profile

Provider Overview

The Provider Overview profile of an API shows a link to the API, a list of access URIs and the API keys.

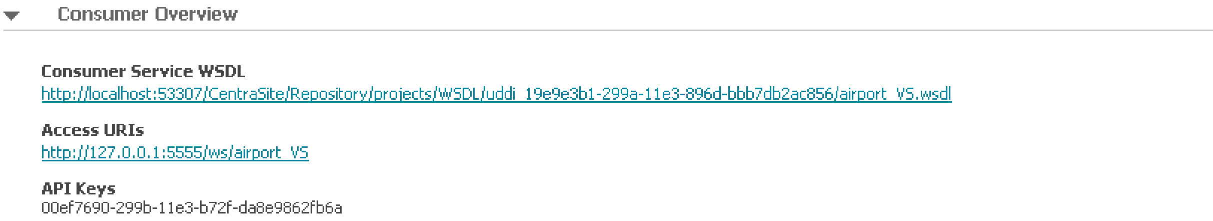
The following example shows the **Provider Overview** profile for an API. Note that the Access URI provides the exact address of each endpoint.

Consumer Overview Profile

The Consumer Overview profile of an API displays the following information:

- For an instance of the SOAP API, this profile displays the Consumer Service WSDL / WSDL URL, and a list of Access URIs and API keys.
- For an instance of the REST-based API, this profile displays a list of Access URIs and API keys.

The following example shows the **Consumer Overview** profile for an API. Note that the Access URI provides the exact address of each endpoint.



Consumer Overview

Consumer Service WSDL
http://localhost:53307/CentraSite/Repository/projects/WSDL/uddi_19e9e3b1-299a-11e3-896d-bbb7db2ac856/airport_VS.wsdl

Access URIs
http://127.0.0.1:5555/ws/airport_VS

API Keys
00ef7690-299b-11e3-b72f-da8e9862fb6a

Identification Profile

In this profile, you specify the precise values for the consumer identifier token(s) that you want to use for identifying consumers for the API. (Alternatively, you may configure this profile to allow unrestricted access.)

For example, if you configure the Identification profile to identify consumers by IP address, the PEP extracts the IP address from a request's HTTP header at run time and searches its list of consumers for the API that is defined by that IP address.

Note: If you want to authenticate consumers, make sure that your policy enforcement point is configured to enable authentication. For information, see the webMethods Mediator documentation or the documentation for your third-party PEP.

Note: For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

In this field...	Do the following...
IPv4 Address	Use this field to identify consumers based on their originating 4-byte IP address range. Specify a range of IPv4 addresses. Type the lowest IP address in the From field and the highest IP address in the To field. For example, "192.168.0.0 and 192.168.0.10" The API will identify only those requests that originate from the specified IP address. If you need to specify additional IP addresses, use the plus button to add more rows.

In this field...	Do the following...
IPv6 Address	Use this field to identify consumers based on their originating 128-bit IPv6 address. Specify a IPv6 address. For example, "fdda:5cc1:23:4::1f" The API will identify only those requests that originate from an IP address that lies between the specified ranges. If you need to specify additional IP addresses, use the plus button to add more rows.
Hostname	Use this field to identify consumers based on a specified host name. Specify the hostname. For example, "pcmachine.ab.com" The API will identify only those requests that originate from the specified host name. If you need to specify additional hostnames, use the plus button to add more rows.
HTTP Token	Use this field to authenticate consumers based on the user name that is transmitted in an HTTP authentication user token. Specify one or more HTTP user names. For example, "SAGUser123" The API will identify only the requests that contain the specified user name encoded and passed in the HTTP authentication user token. If you need to specify additional tokens, use the plus button to add more rows.
WS-Security Token	Use this field to authenticate consumers based on the user name that is transmitted in the SOAP or XML message header (HTTP body). Specify the WSS username token. For example, "userwss" The API will identify only the requests that contain the specified user name passed in the SOAP or XML message header. If you need to specify additional tokens, use the plus button to add more rows.

XPath Token	Use this field to identify consumers based on the result of applying an XPath expression on the SOAP or XML message or request.
	`//*[local-name()= 'Envelope']/* [local-name()='Body']/* ↩`
	`[local-name()= 'echoInt']/* [local-name() ='echoIntInput='] [.='2']`
	The API will identify the requests that contain the XPath and the consumers.
	If you need to specify additional tokens, use the plus button to add more rows.

Consumer Certificate	Use this field to identify consumers based on information in an X.509 v3 certificate. Click **Upload** to locate and select the certificate (.cer) file.
API Key String	*Read-only. String.* The API key for authentication of API consumption. The API key is typically hidden from users who browse the catalog looking for APIs to reuse and is visible only to a particular user (known as the API's owner).
Expiry Date	*Read-only. String.* The expiration date for the API key.

API Key Scope Profile

In this field...	Do the following...
API Service	*Read-only. String*. The name of the an API that is associated with the API key. To view details of the an API, click its hyperlinked name.

5 Setting Permissions on an API

By default, everyone in your organization is permitted to view the APIs that you create. However, only you (as the owner of the API) and users who belong to a role with the "Manage Assets" permission for your organization are allowed to view, edit and delete these API. To enable other users to view, edit and/or delete an API that you have created, you must modify the API's permission settings.

The following sections describe how to set permissions on an API.

Who Can Set Permissions on an API?

When setting permissions on APIs, keep the following points in mind:

- To set permissions on an API, you must belong to a role that has the "Manage Assets" permission or have the Full instance-level permission on the API itself.
- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite *including guest users* (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on an API and group XYZ has Full permission on the same API, users that belong to both groups will, in effect, receive Full permission on the API.

  The same principle applies to users who have both role-based permissions and instance-level permissions on the same API. In this case, users receive the union of the role-based permission and the instance-level permission on the API.
- If you intend to give users in other organizations access to the API, and the API includes supporting documents that you want those users to be able to view, make sure you give those users permission to view the supporting documents as well as the API itself.

Setting Instance Level Permissions on an API Proxy

Use the following procedure to set instance-level permissions on a proxy API using CentraSite Business UI.

To assign permissions to a proxy API

1. In CentraSite Business UI, display the details page for the proxy API whose permissions you want to edit. If you need procedures for this step, see the section **Viewing Details for an API Proxy**.
2. On the API's actions menu, click the **Permissions** 🔑 icon.
3. In the **Assign Permissions** dialog box, select the users or groups to which you want to assign permissions.
4. Use the View, Modify and Full check boxes to assign specific permissions to each user and/or group in the **User/Group Permissions** list as follows:

Permission	Allows the selected user or group to...
View 🔒	View the proxy API.
Modify 🔓	View and edit the proxy API.
Full 🔓	View, edit and delete the proxy API. This permission also allows the selected user or group to assign instance-level permissions to the API.

5. When you assign instance-level permissions on a proxy API, the related objects (for example, bindings, operations, interfaces etc.,) receive the same permissions that are assigned on the API.
6. If you want to ensure that the API's dependent APIs (for example, a WSDL or schema) receive the same permissions, expand the **Advanced Settings** section and mark the checkbox **Propagate permissions**. If you do not mark this checkbox, the permissions of the dependent APIs will not be modified.

   In addition, you can ensure that the dependent APIs of the same object type receive the same profile permissions. To do this, mark the checkbox **Propagate profile permissions**.
7. If at any time, you wish to remove one or more users' or groups' permissions, click the **Delete** 🗑 icon next to the user or group name.
8. Click the **Ok** button to save the permission settings.

9 When you have finished making your changes, click the **Save** icon.

Setting Instance Level Profile Permissions on a Proxy API

Use the following procedure to set instance-level permissions on a proxy API's profiles.

To assign instance-level permissions on a proxy API's profiles

1. Choose the API's **Permissions** action.
2. Locate the user or group for which you wish to set profile permissions. Then click the arrow icon beside the user or group name to open the profile permission list.
3. Use the checkboxes to indicate which profiles the user or group is permitted to view or modify.
4. Click **Ok** to save the new permission settings.
5. When you have finished making your changes, click the **Save** icon.

6 Publishing an API Proxy

This section describes how to publish a proxy API to the webMethods Mediator.

To publish a proxy API, the following prerequisites must be met:

- Before you publish a proxy API to the webmethods Mediator, you must configure the consumption settings for that particular API. For procedures, see *Configuring the API Consumption Settings*.
- You must be a registered user in the CentraSite.
- You must belong to either one of the following roles:
  - “CentraSite Administrator”
  - “API Publisher”

  For more information about roles, see the CentraSite online documentation section *Users, Groups, Roles and Permissions* > *About Roles and Permissions* .
- You must have at least the instance-level Full permission on the proxy API.
- webMethods Mediator must be configured and running on the webMethods Integration Server.
- To expose a proxy API for consumption, make sure that the " Set API Publish Permissions" policy is active. If the policy is in the inactive state, you must activate it. By default, this policy is active.

  The "Set API Publish Permissions" policy includes a "Set Permissions" action that grants the instance-level “View” permission for the proxy API to the users in “Everyone” group. In addition, you can use the option "Propagate permissions to dependent objects". For more information, see the description of the action in the online documentation section *Built-In Design/Change-Time Actions Reference* > *Built-In Actions for Design/Change-Time Policies* .
- The **Publish** action is *not visible* in the details page for a proxy API, unless the following conditions are satisfied:
  - Proxy API should be one of the following asset types: Virtual Service, Virtual XML Service, Virtual REST Service.

- If the proxy API is under the control of an active lifecycle model (LCM), ensure that:
  - Proxy API is in a "deployable" lifecycle state. If you are not certain of what the "deployable" lifecycle state is, consult your CentraSite administrator.
  - Proxy API is associated with a design-time policy that includes the "Change Deployment Status" action and it is set to Yes. This action specifies whether the API is eligible for deployment. For more information, see the description of the action in the online documentation section *Built-In Design/Change-Time Actions Reference > Built-In Actions for Design/Change-Time Policies* .

You can publish a single proxy API or a selected set of proxy APIs. The descriptions in this section give you details on how to do this.

Publishing a Single Proxy API

You use this procedure to publish a single proxy API using its **Publish** action.

To publish a single proxy API

1. Display the details page for the proxy API that you want to publish to webMethods Mediator. For procedures, see the section *Viewing Details for an API Proxy*.
2. On the API detail page, click **Publish** . This opens the **Publish API** dialog.
3. From the **Available Targets** list, choose one or more targets in which you want to publish the proxy API.

   Or:

   If you want to publish the proxy API across all the targets, select the **All Targets** checkbox.

   When attempting to publish a proxy API, without selecting at least on target, a warning popup box appears.
4. In the **Advanced Settings** node, select the **Expose to Consumers** checkbox. This allows unauthorized consumers (guests) to search and access the proxy API.
5. Click **Publish**.

   Or:

   If at any time you wish to terminate this operation, click **Cancel**.
6. A **Publish Progress** popup will display the progress state of publishing the proxy API to the webMethods Mediator.

If the publish process failed, identify and correct the error and then try publishing the proxy API again.

Publishing a Set of Proxy APIs

You use this procedure to publish a set of proxy APIs using the native API's **Virtualize** or **Publish** action.

To publish a set of proxy APIs

1 Display the details page for the native API whose proxy endpoints (APIs) you want to publish to webMethods Mediator. For procedures, see the section *Viewing Details for an API*.

2

On the API details page, click the **Virtualize** action. Go to panel 3 of the **Virtualize** wizard.

Or:

Click the **Publish** action. This opens the **Publish API** dialog.

3 Choose the proxy APIs you want to publish from the drop-down labeled **Virtual Alias**. (The list will contain the proxy APIs that are defined for that particular API.)

4 From the **Available Targets** list, choose one or more targets in which you want to publish the proxy APIs.

Or:

If you want to publish the proxy APIs across all the targets, select the **All Targets** checkbox.

When attempting to publish the proxy APIs, without selecting at least on target, a warning popup box appears.

5 In the **Advanced Settings** node, select the **Expose to Consumers** checkbox. This allows unauthorized consumers (guests) to search and access the proxy APIs.

6 Click **Publish**.

Or:

If at any time you wish to terminate this operation, click **Cancel**.

7 A **Publish Progress** popup will display the progress state of publishing the proxy APIs to the webMethods Mediator.

If the publish process failed, identify and correct the error and then try publishing the proxy APIs again.

7 Managing Consumer Applications

A consumer application is a computer application that consumes (invokes) APIs (services, XML services or REST services) at run time. Typically, consumer applications specify the consumers that are allowed (registered) to consume a particular API. Then, you include the consumer application in the **Consumer** action of the API.

A consumer application in CentraSite is represented by an *Application* asset. The Application asset defines the precise consumer identifiers (for example, a list of user names in HTTP headers, a range of IP addresses, etc.). Thus the policy enforcement point (such as Mediator) can identify or authenticate the consumers that are requesting an API.

You can use Application assets with any supported policy enforcement point (that is, webMethods Mediator or any supported third-party policy enforcement point).

Note: If you want to authenticate consumers (using LDAP or another external authentication mechanism), make sure that your policy enforcement point is configured to enable authentication. For information, see the documentation for your policy enforcement point.

Who Can Create and Manage Consumer Applications?

To create and manage consumer applications, you must belong to a role with the following permissions:

- Create Assets —OR— Manage Assets
- Manage Lifecycle Models (required to change state of consumer applications)

For more information about roles and permissions, see the online documentation section *Users, Groups, Roles and Permissions > About Roles and Permissions* .

Identifying and Authenticating Consumer Applications

To identify and authenticate consumer applications, you perform the following high-level steps:

1. **Include the Evaluate (consumer) action in the API's Message Flow.**
   To identify and authenticate the consumer applications that are requesting a proxy API, that API must have a run-time policy that includes one of the “Evaluate” action. In an “Evaluate” action, you specify the consumer identifier you want to use for identifying and authenticating consumer applications. This action extracts the specified identifier from an incoming request and locates the consumer application defined by that identifier.

   For example, if you configure the Evaluate IP Address action to identify and authenticate consumers, the PEP extracts the IP address from a request’s HTTP header at run time and searches its list of application assets for the application that is defined by that IP address.

You can configure an "Evaluate" action to identify consumer applications based on the appropriate consumer identifier in a request message:

Action Name	Consumer Identifier	Description
Evaluate IP Address	IP Address	The IP address from which the request originated.
Evaluate Hostname	Host Name	The name of the host machine from which the request originated.
Evaluate HTTP Basic Authentication	HTTP Authentication Token	The user ID submitted by the requestor when it was asked to provide basic HTTP credentials (user name and password).
Evaluate WSS Username Token	WS-Security Authentication Token	The WSS username token supplied in the header of the SOAP or XML request that the consumer application submitted to the virtualized service.
Evaluate XPath Expression	Custom Identification	A string produced by applying a specified XPath expression to the SOAP or XML request that the consumer application submitted to the virtualized service.
Evaluate WSS X.509 Certificate	Consumer Certification	The X.509 certificate supplied in the header of the SOAP or XML request that the consumer application submitted to the asset.
Evaluate Client Certificate for SSL Connectivity	Client Certificate for SSL Connectivity	The client's certificate that the consumer application submits to the asset. The certificate is supplied during the SSL handshake over the Transport layer. Communication between the client and the asset must be over HTTPS.

When deciding which "Evaluate" action to use to identify and authenticate a consumer application, consider the following points:

- Whatever identifier you choose to identify a consumer application, it must be unique to the application. Identifiers that represent user names are often not suitable because the identified users might submit requests for multiple applications.
- Identifying applications by IP address or host name is often a suitable choice, however, it does create a dependency on the network infrastructure. If a consumer application moves to a new machine, or its IP address changes, you must update the identifiers in the application asset.
- Using X.509 certificates or a custom token that is extracted from the SOAP or XML message itself (using an XPATH expression), is often the most trouble-free way to identify a consumer application.

For more information about the "Evaluate" actions, see the online documentation section *Run-Time Governance Reference > Built-In Run-Time Actions Reference for APIs*. Additionally, within that section see *Usage Cases for Identifying/Authenticating Consumers*.

2. **Create an application asset in the registry.**
   In the application asset you specify precise values for the consumer identifier(s) that you specified in the “Evaluate” action. For details, see *Creating a Consumer Application Asset*.
3. **Specify the application asset in the Consume action of the API to be consumed.**
   The Consume action is located in the API's detail page.

The run-time behavior of identifying and authenticating consumers is as follows:

1. CentraSite translates the application asset to the appropriate WS-Security policy assertions or an equivalent XML when the application asset is enforced by the PEP.
2. When a consumer application requests access to an asset, the PEP tries to map the consumer's identifier (which is found in the request) to an identifier in the application asset.
   - If the identifier is an IP address, a host name, a custom identification string or a consumer certificate, the PEP tries to identify the consumer (the consumer is not authenticated).
   - If the identifier is an HTTP Authentication token or a WS-Security Authentication token, the PEP tries to *authenticate* the consumer. If you use webMethods Mediator, authentication is handled by LDAP or by another external authentication mechanism, depending on how Mediator is configured. If you use a third-party PEP, authentication capabilities depend on the PEP.
3. The identified or authenticated consumer information is published back to the registry as part of the transaction or other events. This information is used to correlate the consumer-specific run-time dependencies.

Creating a Consumer Application

Use the following procedure to create a consumer application asset.

To add an asset to the catalog

1 In CentraSite Business user interface, click the **Create Assets** activity. This opens the **Create Asset** wizard.

2 In **Basic Information** panel, enter values for the following fields:

In this field...	Specify...
Name	A name for the application asset. An asset name can contain any character (including spaces).
Type	The **Application** asset type.
Organization	The organization to which the application asset belongs.

In this field...	Specify...

Initial Version	*Optional*. An identifier for the initial version of the application asset. The default is "1.0", but you can use any versioning scheme you choose. The version identifier does not need to be numeric.
	Examples:
	`0.0a`
	`1.0.0 (beta)`
	`Pre-release 001`
	`V1-2007.04.30`
	You can later create new versions of the application asset (see the section *Versioning an Asset* in the document *Managing Assets*).

Description	*Optional*. A comment or descriptive information about the new application asset.

3 Click **Next**.

Note: If you do not enter information into a required field, the system will alert you with a red error icon. Pointing to the icon shows a hint with an error description.

4 In the **Preview** panel, click **Finish**.

5 Configure the profiles of the detail page, as described in the following sub-sections.

6 If you belong to a role that has the "Register as Consumer" permission, the entry **Register as Consumer** is enabled in the **Actions** menu of the details page. If you select this menu entry, a dialog opens that lets you select users, groups and consumer applications that can use this asset. The request must be subsequently approved or rejected by the owner of the asset.

7 Specify the application asset in the Consume action of the API to be consumed. To do this, open the detail page of the API to be consumed and specify the application asset in the **Consume** action.

Editing a Consumer Application

Use the following procedure to edit the attributes associated with a consumer application asset in the Business UI.

When editing attributes, keep the following general points in mind:

- If you are not the owner of the asset, you cannot edit the asset unless you have Modify permission on the asset (granted though either a role-based permission or an instance-level permission).
- When you view the details for the asset, you will only see profiles for which you have View permission. You will only be able to edit the profiles on which you have Modify permission.

To edit the attributes for a consumer application asset

1. In CentraSite Business UI, *Browse* or *Search* to display the application asset(s). For procedures on how to browse or search the CentraSite, see the section *Browsing the CentraSite Catalog* or *Searching the CentraSite Catalog* in the document *Managing the CentraSite Catalog*.
2. Locate the application asset whose details you want to view.
3. On the asset's actions menu, click the **Edit** icon.
4. Locate the application asset whose details you want to view and, from its context menu, select **Details**.
5. To edit the application asset's **Name**, **Description** or user-defined version number, place the cursor in the appropriate field and modify the text as required.
6. To modify the extended attributes associated with the asset, do the following:

    1. Select the profile that contains the attribute(s) that you want to modify.
    2. Edit the attributes on the profile as necessary.

7. When you have finished making your edits, click the **Save** icon from the actions menu.

    When you are prompted to confirm the save operation, click **Yes**.

    Or:

    If at any time you want to abandon your unsaved edits, click **Close**. CentraSite will ask you if you want to save your edits. Click **No** to abandon your edits and return the asset's attributes to their previous settings.

Configuring the Consumer Identification Profile

You use the Identification profile to specify the precise values for the consumer identifier(s) that you specified in the "Evaluate" action.

Notes:

1. If you specify *multiple* identifiers, the system evaluates them with the identifier defined in the "Evaluate" action.
2. If you want to authenticate consumers, make sure that your policy enforcement point is configured to enable authentication. For information, see the webMethods Mediator documentation or the documentation for your third-party PEP.

To configure the consumer identifiers

- Specify values for one or more consumer identifier tokens.

  Note: The value(s) that you specify in the Identification profile depend on how the run-time policy's "Evaluate" is configured. For example, if "Evaluate IP Address" action is configured to identify and validate consumers by their IP address, you should specify the consumer IP addresses here. For information about this action, see the online documentation section *Run-Time Governance Reference > Built-In Run-Time Actions Reference for APIs*.

Note: For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

In this field...	Do the following...
IPv4 Address	Specify a range of IPv4 addresses. Type the lowest IP address in the **From** field and the highest IP address in the **To** field. This will identify only those requests originating from any IP address that lies between the specified range. Example: "192.168.0.0 and 192.168.0.10" If you need to specify additional IP addresses, use the plus button to add more rows.
IPv6 Address	Specify a IPv6 address. This will identify only those requests that originate from the specified IP address. Example: "fdda:5cc1:23:4::1f" If you need to specify additional IP addresses, use the plus button to add more rows.
Hostname	Specify the hostname. This will identify only those requests that originate from the specified hostname. Example: "pcmachine.ab.com" If you need to specify additional hostnames, use the plus button to add more rows.
HTTP Authentication Token	Specify one or more HTTP user names. This will identify only those requests that contain the specified user names encoded and passed in the HTTP authentication user token. Example: "SAGUser123" If you need to specify additional tokens, use the plus button to add more rows.
WS-Security Authentication Token	Specify the WSS username token. This will identify only those requests that contain the specified user name passed in the SOAP or XML message header. Example: "userwss" If you need to specify additional tokens, use the plus button to add more rows.
XPath Token	Specify one or more XPath expressions. This will identify only those requests that contain the specified XPath in the SOAP or XML message or request. Example:

In this field...	Do the following...

	`//*[local-name()= 'Envelope']/* [local-name()='Body']/* ↩`
	`[local-name()= 'echoInt']/* [local-name() ='echoIntInput='] ↩`
	`[.='2']`
	If you need to specify additional tokens, use the plus button to add more rows.

Consumer Certificate	Specify the X.509 certificates that help the API owner to identify requests from you. Click **Upload** to locate and select the certificate (.cer) file.

Publishing a Consumer Application

You can publish consumer application assets to a policy enforcement point (such as webMethods Mediator) in either of the following ways:

- **From the API Details page in CentraSite Business UI**

  You can publish multiple consumer applications to a Mediator target in a single step (see ***Publishing Consumer Applications Using the CentraSite Business UI***).

- **From the Operations > Deployment page in CentraSite Control**

  You can publish multiple consumer applications to a Mediator target in a single step (see *Publishing Consumer Applications Using the CentraSite Control*).

- **Running a script file from a command line**

  You can run a script file to publish multiple consumer applications to a Mediator target in a single step (see *Publishing Consumer Applications Using the Command Line Tool*).

Publishing Consumer Applications from the API Details Page

Note that the CentraSite Business UI does not have a dedicated user interface available to publish consumer applications to webMethods Mediator. Instead, deploys the consumer applications that are associated to an API when that particular API is published to the Mediator.

You use this procedure to publish the consumer applications associated with an API using its **Publish** action.

▶ To publish consumer applications

1 Display the details page for the API whose associated consumer applications you want to publish to webMethods Mediator. For procedures, see the section ***Viewing Details for an API Proxy***.

2 On the API detail page, click **Publish** . This opens the **Publish API** dialog.

3 From the **Available Targets** list, choose one or more targets in which you want to publish the consumer applications.

Or:

If you want to publish the consumer applications across all the targets, select the **All Targets** checkbox.

When attempting to publish the consumer applications, without selecting at least on target, a warning popup box appears.

4 In the **Advanced Settings** node, select the **Expose to Consumers** checkbox. This allows unauthorized consumers (guests) to search and access the consumer applications.

5 Click **Publish**.

Or:

If at any time you wish to terminate this operation, click **Cancel**.

6 A **Publish Progress** popup will display the progress state of publishing the consumer applications to the webMethods Mediator.

If the publish process failed, identify and correct the error and then try publishing the consumer applications again.

Deleting a Consumer Application

Use the following procedure to delete a consumer application asset.

To delete an application asset

1 In CentraSite Control, display the detail page of the application asset that you want to delete. If you need procedures for this step, see the section *Administering the CentraSite Business UI* > *Managing Assets* > *Viewing Details for an Asset* .

2 Locate the application asset that you want to delete.

3 On the asset's actions menu, click the **Delete** icon.

4 When you are prompted to confirm the delete operation, click **Yes**.

The asset is permanently removed from the CentraSite registry.

You can delete multiple application assets in a single step.

▶ To delete multiple application assets in a single operation

1. In CentraSite Business UI, *Browse* or *Search* to display the application asset(s). For procedures on how to browse or search the CentraSite, see the section *Browsing the CentraSite Catalog* or *Searching the CentraSite Catalog* in the document *Managing the CentraSite Catalog*.
2. Mark the checkbox of each application asset you want to delete.
3. In the **Actions** menu, click **Delete**.

   ⚠ **Important:** If you have selected several application assets where one or more of them are predefined application assets, you can use the **Delete** button to delete the assets. However, as you are not allowed to delete predefined application assets, only assets you have permission for will be deleted. The same applies to any other application assets for which you do not have the required permission.

8 Registering as Consumers of an API

Clients that need to call (consume) APIs must register with CentraSite as consumers of the API.

Clients can register as consumers of APIs as a:

- User or user group with a valid CentraSite user account. For procedures, see ***Authorized Centrasite User Accessing API as Logged on User***.
- Guest user (with or without a valid CentraSite user account). For procedures, see ***Authorized CentraSite User Accessing API as Guest***or ***Unauthorized User Accessing API as Guest***.
- Consumer application (which is represented as an **Application asset**). An Application asset defines precise consumer identifiers (for example, a list of user names in HTTP headers, a range of IP addresses, etc.). Thus Mediator can identify or authenticate the consumers that are requesting an API. For procedures, see ***Registering Application Assets as Consumers***.

The API provider (owner of the API) enforces the type of authentication (API key or OAuth2 token) required for consuming an API. Based on the authentication enforced for the API, an API consumer will request the API key or the OAuth2 token in order to call (consume) that API.

- Clients that want to use the API key to call (consume) an API in CentraSite must:

  1. Register as a consumer for the API.

     When the client registration request is approved, the client receives an API key (a base64-encoded string of the `consumer-key:consumer-secret` combination). It works for both SOAP and REST calls.

  2. To call the API, the client must pass the API key in the HTTP request header or as a query string parameter. The use of this key establishes the client's identity and authentication.

     For information about using API keys to consume APIs, see ***Using Your API Keys for Consumption***.

- The type of OAuth2 authorization grant that Mediator supports is “Client Credentials”. Client credentials are used as an authorization grant when the client is requesting access to protected resources based on an authorization previously arranged with the authorization server. That is, the client application gains authorization when it registers with CentraSite as a consumer.

  Clients that want to use the OAuth 2.0 protocol to call (consume) APIs in CentraSite must:

  1. Register as a consumer for the API.

     When the client registration request is approved, the client receives client credentials (a client_id and client_secret).

  2. Pass the client credentials to the Mediator-hosted REST service mediator.oauth2.getOAuth2AccessToken.

     This service will provide an OAuth2 access token to the client. For information about this service, see ***Fetching and Using Your OAuth2 Access Tokens for Consumption***.

3. To call the API, the client must pass their OAuth access token as an integral part of the HTTP request header.

   An OAuth2 token is a unique token that a client uses to invoke APIs using the OAuth 2.0 protocol. The token contains an identifier that uniquely identifies the client. The use of a token establishes the client's identity, and is used for both the authentication and authorization.

This section covers the following topics:

Registering Users as Consumers

Users can register themselves as consumers of specified APIs, using the Consume action. That is, users can request permission to access specified APIs in the registry. The owners of the APIs may approve or reject such requests. The Consume action applies only to proxy APIs.

Note: To enable CentraSite to issue email messages, an administrator must first configure CentraSite's email server settings. For procedures, see the section *Configuring the Email Server* in the document *Basic Operations*.

To register a user as a consumer for an API

1 In CentraSite Business UI, display the details page for the API that you want to consume. For procedures, see the section ***Viewing Details for an API Proxy***.

2 On the API details page, click **Consume** (🧺). This opens the **Consume API** dialog.

3 Depending on the type of user account you have in CentraSite, you must complete one of the following procedures:

Authorized Centrasite User Accessing API as a Logged-On User

- OR -

Authorized CentraSite User Accessing API as a Guest

- OR -

Unauthorized User Accessing API as a Guest

Authorized Centrasite User Accessing API as Logged-On User

You must at least have the instance-level View permission for the specified API. If your user account belongs to a role that has either the "Manage Assets", "Create Assets", "Modify Assets" or "View Assets" permission for an organization, you automatically have permission to register as consumer for all APIs in that particular organization.

1. Specify the **API Key** or **OAuth2** client credentials.
2. In the **Consumer Name** field, specify your CentraSite username.
3. Select the **Email me** checkbox in order to receive auto-generated workflow notifications, and then specify your email address.
4. Enter a reason to request the API for consumption.
5. If the API's policy governance rule includes one or more "Evalaute" actions, you will see the **Consumer Identifier** field. Enter your consumer identifier, by which the provider will recognize your messages at run time. For details, see *Configuring the Consumer Identification Profile*.
6. Click **Consume**.

   A request is sent to the designated approvers. Upon approval, a request for consumption of the selected API will be sent to the provider of the API, who will then generate the API key / OAuth2 client credentials.

   Once approved, the API consumption request will be processed and a notification will be sent to you at the specified email address.

   If an approval workflow is not defined for the API, the API key / OAuth2 credentials is generated immediately.

   To get your generated API keys or OAuth access tokens for consumption, refer to the section *Obtaining Your API Keys and Access Tokens for Consumption*.

Authorized CentraSite User Accessing API as a Guest

1. In the Login page, enter your username and password and click **Next**.
2. Specify the **API Key** or **OAuth2** client credentials.
3. In the **Consumer Name** field, specify your CentraSite username.
4. Select the **Email me** checkbox in order to receive auto-generated workflow notifications, and then specify your email address.
5. Enter a reason to request the API for consumption.
6. If the API's policy governance rule includes one or more "Evalaute" actions, you will see the **Consumer Identifier** field. Enter your consumer identifier, by which the provider will recognize your messages at run time. For details, see *Configuring the Consumer Identification Profile*.
7. Click **Consume**.

A request is sent to the designated approvers. Upon approval, a request for consumption of the selected API will be sent to the provider of the API, who will then generate the API key / OAuth2 credentials.

Once approved, the API consumption request will be processed and a notification will be sent to you at the specified email address.

If an approval workflow is not defined for the API, the API key / OAuth2 credentials is generated.

To get your generated API keys or OAuth access tokens for consumption, refer to the section ***Obtaining Your API Keys and Access Tokens for Consumption***.

Unauthorized User Accessing API as a Guest

If you are a guest user without a valid CentraSite user account, CentraSite internally executes a consumer onboarding workflow. This workflow helps you to onboard in an organization of interest within the CentraSite registry/repository. An onboarding request is sent to the organization’s administrator for approval. On successful onboarding of the user, a request for consumption of the selected API will be sent to the provider of the API who will generate the API key or OAuth2 client credentials.

1. In the Request an Account page, specify the following:
   a. Enter your **First Name** and **Last Name**.
   b. Type in your password in the **Password** field.
   c. Retype the password in the **Confirm Password** field.
   d. Enter the **Email** address which you will use as username when signing into CentraSite Business UI.
   e. Enter the **Organization** you want to join.

      If the **Organization** field is left blank, CentraSite will automatically register the user as a consumer in the organization that was configured in the Global Onboarding Policy.
   f. Click **Next**.
2. Specify the **API Key** or **OAuth2** client credentials.
3. In the **Consumer Name** field, specify your CentraSite user name.
4. Select the **Email me** checkbox in order to receive auto-generated workflow notifications, and then specify your email address.
5. Enter a reason to request the API for consumption.
6. If the API's policy governance rule includes one or more "Evalaute" actions, you will see the **Consumer Identifier** field. Enter your consumer identifier, by which the provider will recognize your messages at run time. For details, see ***Configuring the Consumer Identification Profile***.
7. Click the **Consume** button.

A consumer registration request is sent to the organization's administrator for approval. Upon successful registration of the consumer, a request for consumption of the selected API will be sent to the provider of the API, who will then generate the API key / OAuth2 credentials.

If an approval workflow is not defined for the API, the API key / OAuth2 credentials is generated immediately.

To get your generated API keys or OAuth access tokens for consumption, refer to the section *Obtaining Your API Keys and Access Tokens for Consumption*.

Registering Application Assets as Consumers

If you have permissions to view an API, and you belong to a role that includes the "Register as Consumer" permission, the Consume action is enabled in the API details page. This action opens a dialog that lets you request the right to be a consumer of the specified API. You can request the right for any consumer application owned by any organization.

The request must be subsequently approved or rejected by the owner of the API.

▶ To register an application asset as a consumer for an API

1 Display the details page for the API you want to consume. If you need procedures for this step, see the section *Viewing Details for an API Proxy*.

2 On the API's actions menu, click the **Consume** icon. This opens the **Consume API** dialog.

3 In the **Application** textbox, type the keyword(s) to search for. CentraSite applies the filter to the application asset's name. Choose an application asset from the selection list.

4 If you want to specify additional application assets, use the plus button beside the **Application** field to create a new **Application** input field, and choose another application asset.

5 When you have specified all required applications, click **Consume**. Requests to register the applications are sent to the owner of the specified API.

6 The owner of an API can either accept or decline a "Register as Consumer" request as follows:

- Go to the API details page.
- You will see the pending consumer registration requests () for an API in the description area of the **Basic Information** profile, for example, "`N consumer registration requests are pending`".

  If there are no pending consumer registration requests for the API, this is displayed as "`0`".

- Click the hyperlinked number (“N”) to open the **Pending Consumer Registration Requests** dialog. This dialog contains a list of all consumer registration requests that have been submitted for the particular API, including requests that were auto-approved.
- Choose the consumer registration request that you want to review and approve by clicking its hyperlinked name.

  The details for the request will appear in the **Consumer Registration Request** dialog.
- In the **Comment** text box, type a comment (e.g., *"Request rejected. Add required specifications to this API and resubmit"*.).
- Click the **Accept** or **Reject** button to approve or reject the request.

  After the applications are approved to consume the specified API, CentraSite automatically changes the consumer count in the API's **Basic Information** profile.

  For details, see the section *Monitoring Consumer Count for an Asset*.

Viewing Consumer Registration Requests

To view a summary of all "Register as Consumer" requests, go to the API details page:

- If you are the owner of an API, and another user has made a request to register as a consumer of the API, you can view the request here. As the API owner, you can accept or decline the request.
- If you have made a request to register as a consumer of an API owned by another user, you can view the status of the request here.

Note: If an API has a pending state change approval request and a pending consumer registration request, then the pending state change approval takes priority over the pending consumer registration request.

Monitoring Consumer Count for an API

CentraSite Business UI has extensive support for consumer-provider tracking that allows you to monitor the number of consumers for an API.

The number of users who consume an API is displayed in brackets with icons (representing the *Consumers*) in the description area of the **Basic Information** profile in the API details page, for example, "(5) Consumers". If no consumers are registered for the API, this is displayed as "(0) Consumers".

Clicking on this consumer count displays the consumers' information.

9 Obtaining Your API Keys and Access Tokens for Consumption

The following section describes how to fetch your API keys and access tokens that enable you to consume APIs.

Note: To enable CentraSite to issue email messages, an administrator must first configure CentraSite's email server settings. For procedures, see the section *Configuring the Email Server* in the document *Basic Operations*.

Viewing Details of Your API Keys

If you will be the using the API key authentication and you have successfully **registered as a consumer for an API**, you should have received your API key details.

The CentraSite Business user interface enables users to view details of an API key in the following ways:

- Through the email notification messages that were auto-generated by CentraSite. For procedures, see the section ***Using the Email Notifications***.
- Through the Consumer Overview profile of an API. For procedures, see the section ***Using the API Details Page***.
- Through the User Preferences page for a consumer. For procedures, see the section ***Using the User Preferences***

Using the Email Notifications

Once your API registration request is approved, CentraSite sends an automated email message containing details of the API key value and its usage to both the approver and consumer.

If you want to receive email messages of the API key, make sure:

- You have the notification option set as **Email** in the User Preferences page.
- You have specified a valid email address.

Access your mailbox.

The information contained in the email message depends on whether you access the API as a consumer or the owner.

- If you were the requestor for the API registration, you will see the following information:
  - API key
  - Key expiration date
  - API key usage
- If you are the owner of the API, you will see the following information:

- API key
- Key expiration date

Using the API Details Page

You can use the following procedure to view the details of an API key via the API Details page.

To view the details of an API key

1 In CentraSite Business UI, display the details page of the API whose key value want to view. For procedure, see the section *Viewing Details for an API Proxy*.

2 In the API Details page, go to **Advanced Information -> Consumer Overview** profile.

The information contained in this profile depends on whether you access the API as a consumer or the owner.

- If you were the requestor for API registration, you will see the following information:
  - API key
  - Key expiration date
  - API key usage
- If you are the owner of the API, you will see the following information:
  - API key
  - Key expiration date

Using the User Preferences

Alternatively, you can use the following procedure to view the details of all your API keys via the User Preferences page.

To view the details of an API key

1 In the CentraSite Business UI navigation bar, click the profile name.

The User Preferences page appears.

2 Display **My Access Keys**.

This shows a list of your API keys in CentraSite.

Note: The My Access Keys panel IS NOT VISIBLE if at least one API key is not available in your CentraSite registry/repository.

3 Each entry in the API key list contains the following information:

- API key
- Key value
- Key expiration date

Additionally, you can do the following operations on the API key:

- Renew ⟳ your API key, provided the key has a limited usage period, as described in the section *Renewing Your API Key*.
- Revoke ⟳ your API key temporarily from the CentraSite registry, as described in the section *Revoking Your API Key*.
- Delete 🗑 your API key permanently from the CentraSite registry, as described in the section *Deleting Your API Key*.

Using Your API Keys for Consumption

RESTful APIs are often exposed over the open Internet for consumption. API providers need a mechanism to prevent unauthorized access to the API. One approach is to provision consumers with API keys. Those keys can be used as authentication tokens.

CentraSite API Management Solutions automatically generates API keys when consumers request to consume APIs. The API providers can view, approve and set expiration for API keys. This ensures that no consumer can access a protected API without a valid key.

API keys are verified at runtime to ensure that:

- The API key presented is valid and has not expired.
- The API key presented is approved to consume an API that includes the URI in the request.

Note: Throughout this section, the term “Virtual Alias” is also referred to as “API” or simply “API”. The name can be used interchangeably.

This section provides information about consuming APIs that are published in the webMethods Mediator. If you are using a different kind of PEP, refer to its documentation for publishing procedures and information.

- What Happens When You Request for API Consumption?
- What Happens When Consumption Fails?

The API Consumption Model

To enable a consumer to consume an API, the following events must occur:

1. The consumer sends a request to consume a specified API. The request must include the consumer's authentication credentials.
2. CentraSite generates the API key for consumption of the API (the specific key generation steps depend on the configuration settings defined by the Provider of that particular API). Later, CentraSite prepares the API for publishing and invokes the API Key Generation policy on the Mediator. The consumer will use this API key in order to consume this API.
3. The API Key Generation policy publishes the API in the Mediator.
4. If the publication is successful, the API Key Generation policy returns a success message, including data that is pertinent to the published API. (This includes the API key that is required for consuming this API). If the publishing is unsuccessful, the deployer service returns a failure message.
5. The consumer accesses the URL for API consumption, sends the API key as integral part of the HTTP request header or as a query string, and upon validation of the API key consumes the API.
6. If the consumption is successful, the consumer uses the API. If the consumption is unsuccessful for some reasons of authentication, a 500 fault is returned.

How Does Mediator Evaluate Consumers at Run Time?

To determine the consumer from which an API consumption request was submitted, an API must have the policy governance rule defined with at least one Evaluate action. This action extracts a specified evaluator from an incoming request and locates the consumer defined by that identifier.

For example, if you configure the "Evaluate IP Address" action to evaluate consumers by IP address, Mediator extracts the IP address from a request's HTTP header and searches its list of consumers for the consumer that is defined by that IP address.

You can configure the Evaluate actions to evaluate consumers based on the following information in a request message.

Evaluator	Description
IP Address	The IP address from which the request originated.
Host Name	The name of the host machine from which the request originated.
HTTP Authentication Token	The user ID submitted by the requestor when it was asked to provide basic HTTP credentials (username and password).
WS-Security Authentication Token	The WSS username token supplied in the header of the SOAP request that the consumer submitted to the API.
XPath	The custom authentication credentials (tokens, or username and password token combination) supplied in the HTTP request header that the consumer submitted to the API.
Consumer Certificate	The X.509 certificate supplied in the header of the SOAP request that the consumer submitted to the API.

How Does a Consumer Use the Generated API Key?

REST services rely on HTTP methods like GET, POST, PUT and DELETE to make request to an API provider, so the API keys are closely tied to these HTTP methods, where they are sent as part of these HTTP method requests.

CentraSite allows you to set API keys as part of the HTTP header or as the query component of an API request.

⚠ **Important:** In the case where a consumer is sending a request with both credentials (HTTP header) and (query string), the HTTP header take precedence over the query string when the Mediator is determining which credentials it should use for the consumption.

- Request Header
- Query String

Request Header

The API keys are passed as the HTTP header component of an API consumption request. The HTTP header corresponds to an array of header names to include for that particular API consumption.

The following example demonstrates a typical HTTP request with API keys that form the header value of the API Access URL.

```
x-CentraSite-APIKey:a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

Query String

The API keys are passed as the query component of an API consumption request.

The following example demonstrates a typical HTTP GET request with API keys that form a query string of the API Access URL.

```
http://localhost:5555/ws/RestAPI?APIKey=a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

Notice that the API keys are added to the path after a “?”, and specified as key-value pair.

What Happens When You Request for API Consumption?

When you request an API for consumption using the Access URL and the generated API key, CentraSite automatically validates the APIs run-time actions to ensure that any "Evaluate" action that appears in the policy governance rule of an API is validated with the consumer requesting for that API.

What Happens When Consumption Fails?

If the API consumption encounters a problem due to one or more of the following reasons, a 500 SOAP fault is returned.

- If the API key value in the HTTP header or the query string is authenticated as invalid.

  The sample message looks like this:

  ```
  The request  is authenticated as invalid.
  ```

- If the HTTP header is not present in the request.

  The sample message looks like this:

  ```
  A required header is missing in the request.
  ```

- If the API key value in the HTTP header is expired.

  The sample message looks like this:

```
The API key has expired.
```

Fetching and Using Your OAuth2 Access Tokens for Consumption

If you will be using the OAuth 2.0 protocol and you have successfully **registered as a consumer for an API**, you should have received your OAuth2 client credentials (a client_id and client_secret).

Now you need to obtain an OAuth2 access token by passing your client credentials to the Mediator-hosted REST service mediator.oauth2.getOAuth2AccessToken. This service will provide an OAuth2 access token that you can subsequently include in your requests to call the API.

The service's input parameters are:

- `client_id`
- `client_secret`
- `scope` (optional). The scope value is the name of the virtual service. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed, and adds the scope to the response. To pass the scope, pass it in the request body.

Ways for Clients to Provide the Inputs

There are three ways in which a client can provide the inputs for this service:

- Provide inputs in the Basic authentication header (recommended).

  The client can provide the client credentials (client_id and client_secret) in the Authorization header using the following form:

  ```
  Authorization: Basic <base-64-encoded client_id:password, client_secret>
  ```

  If you want to pass the scope, pass it in the request body.

- Provide JSON inputs for the service.

  The client can send a JSON request to the service in the following form:

```
{
"client_id" : "",
"client_secret": "",
"scope" : ""
}
```

Note: The client should contain the header Content-type:application/json in the request.

- Provide inputs in the request body

  The OAuth2 specifications do not support sending the client credentials over the URL as URL-Encoded. However, you can send the client credentials in the request body using the following form:

  ```
  client_id=<client_id>&client_secret=<client_secret>&scope=<scope>
  ```

  Note: The client should contain the header `Content-type:application/x-www-formurlencoded` in the request.

Note: If a client provides the client_id and client_secret in both the Authorization header and the request body, the credentials given in the Authorization header are used.

Using HTTPS for Granting Access Tokens

For security reasons it is recommended to use HTTPS in your production environment. If you will be using HTTPS as the transport protocol over which the OAuth2 access tokens will be granted authorization, you must set the parameters pg.oauth2.isHTTPS and pg.oauth2.ports as described in the section *Advanced Settings* in the document *Administering webMethods Mediator*.

Responses Returned to Clients

Following are sample responses that are returned to the client:

- Sample XML response:

  ```
  <Response
  xmlns="https://localhost/rest/pub.mediator.oauth2.getOAuth2AccessToken">
  <access_token>db95b40095f31439a1cd8f411e64abe8</access_token>
  <expires_in>3600</expires_in>
  <token_type>Bearer</token_type>
  </Response>
  ```

- Sample JSON response:

```
{
"access_token": "db95b40095f31439a1cd8f411e64abe8",
"token_type": "Bearer",
"expires_in": 3600
}
```

10 Managing Your API Keys

An API key is a secret code that you can use to identify yourself to CentraSite when you interact with API. You generate an API Key in by registering as consumer for the API, and then use the key in interactions with the specific API published to Mediator. Multiple interactions may be performed with the same API key.

The following sections describe the various operations (renew, revoke and delete) you can perform on the API key at the disposal of the API provider (owner of the API).

Renewing Your API Key

API keys have an expiration period, which is set by the API provider. After an API key is generated, sometimes the API consumer might have to renew the old key due to expiration or security concerns. The API provider can also change the expiration period for the API key or set it so that the key never expires. For more information, see the section *Configuring the API Consumption Settings for API Key Authentication*.

To request for renewal of an API key, the following prerequisites must be met:

- API provider must have configured the predefined policy `API Key Renewal`, which enables the CentraSite Administrator, API Provider or designated approvers to approve or reject the "Renew API Key" requests. For information, see the section *API Key Renewal Policy*.
- A target instance (for example, Mediator) should be up and running. For information on targets, see the section *Run-Time Targets*.
- The consumer must be a registered consumer for the specified API. For more information on registering as consumers for an API, see ***Registering as a Consumer for an API***.

To renew your API key

1. Log in to the CentraSite Business UI
2. Click the profile name in the navigation bar (at the top right of any page).

   This displays the User Preferences page.
3. Display **My API Keys**.

   This shows a list of all your API keys in CentraSite.
4. Click the **Renew** ⟳ icon to the right of the API key you wish to renew.

   Important: If the API key has an unlimited expiration period, the **Renew** icon is *NOT* visible in the user interface.

Sometimes you might have to require an approval to renew the API key. If your API has the **Require Approval** configured in the API Consumption Settings, CentraSite will not renew the API key until the required approvals are obtained. However, if an approval workflow is not configured for the API, the API key is renewed immediately. For more information about approval actions, see *Working with Approval Workflows*.

After renewing the API key, CentraSite automatically publishes the API to the Mediator, triggered by a *Deploy Access Key* action that is included in the *API Key Renewal* policy.

Once the API key renewal request is approved by the designated approvers, CentraSite sends an email message to the API consumer informing the new validity of API key.

Revoking Your API Key

The API consumer might want to revoke an API key if, for example, the key is no longer needed or if an error is found in the API.

To request for revocation of an API key, the following prerequisites must be met:

- API provider must have configured the predefined policy `API Key Revocation`, which enables the CentraSite Administrator, API Provider or designated approvers to approve or reject the “Renew API Key” requests. For information, see the section *API Key Revocation Policy*.
- A target instance (for example, Mediator) should be up and running. For information on targets, see the section *Run-Time Targets*.

To revoke your API key

1. Log in to the CentraSite Business UI.
2. Click the profile name in the navigation bar (at the top right of any page).

   This displays the User Preferences page.
3. Display **My Access Keys**.

   This shows a list of all of your API keys in CentraSite.
4. Click the **Revoke** 🗑 icon to the right of the API key you wish to revoke.

   A confirmation message appears that the API key will be revoked.

Once the API key revocation is processed, CentraSite sends an email message to the API Consumer informing the request has been processed successfully.

For information about the email notifications for API key revocation, see the section *Configuring the email notification for key revocation*.

Deleting Your API Key

The API key consumer can delete API keys. Deleting an API key permanently removes the key from the CentraSite registry/repository. Deleting an API key will not remove the API that is associated with it.

When you delete an API key, CentraSite removes an entry for the API key (that is, it removes the instance of the API key from CentraSite's object database). Also note that:

- An API key can only be deleted if it is already revoked.
- You cannot delete an API key that is in the “pending” mode (example, awaiting a renew approval).
- You must be a registered consumer for the specified API. For more information on registering as consumers for an API, see ***Registering as Consumers for an API***.

To delete your API key

1 In CentraSite Business UI, display the details page for the API key that you want to delete. If you need procedures for this step, see the section ***Viewing the Details for an API***.

2 On the API key’s actions menu, click **Delete** (🗑).

3 When you are prompted to confirm the delete operation, click **Yes**.

The API key is permanently removed from the CentraSite registry.

You can delete multiple API keys in a single step. The rules described above for deleting a single API key apply also when deleting multiple API keys.

⚠ **Important:** If you have selected several API keys where one or more of them are not already revoked, you can use the **Delete** button to delete the keys. However, as you are not allowed to delete unrevoked keys, only key you have revoked will be deleted.

To delete multiple API keys in a single operation

1 In CentraSite Business UI, use either the Browse or Search feature to select a set of API keys you want to delete. If you need information on how to browse or search the CentraSite catalog, refer to the section *Browsing the CentraSite Catalog* or *Searching the CentraSite Catalog* in the document *Managing the CentraSite Catalog*.

2 Mark the checkbox next to the name of each API key you want to delete.

3 In the actions menu, click **Delete** (🗑).

Note: If one or more or the selected APIs is in pending mode (example, awaiting approval), an error message will appear and no API keys will be deleted.

11 Displaying Runtime Information for an API Proxy

You can view the events and metrics for an API in its **Runtime Metrics** profile and **Runtime Events** profile.

To view the runtime information for an API, you must have View permission on the **Runtime Metrics** profile and **Runtime Events** profile of that particular API.

The following sections describe how to view the runtime information of an API.

The Runtime Metrics

You view the following types of Key Performance Indicator (KPI) metrics that webMethods Mediator creates in the Runtime Metrics profile of each API.

Metric	Description
Availability	The percentage of time that a service was available during the current interval. A value of 100 indicates that the API was always available. If invocations fail due to policy violations, this parameter could still be as high as 100.
Success Count	The number of successful service invocations in the current interval.
Total Request Count	The total number of requests (successful and unsuccessful) in the current interval.
Fault Count	The number of failed invocations in the current interval.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. This is measured from the moment Mediator receives the request until the moment it returns the response to the caller.
Minimum Response Time	The minimum amount of time it took the service to complete an invocation in the current interval.
Maximum Response Time	The maximum amount of time it took the service to complete an invocation in the current interval.

Notes:

1. For more information about KPI metrics, see *Key Performance Indicator Metrics* in the document *Administering webMethods Mediator*
2. For details about intervals, see *The Metrics Tracking Interval* in the document *Administering webMethods Mediator*.
3. By default, the Response Time metrics do not include metrics for failed invocations. To include metrics for failed invocations, set the `pg.PgMetricsFormatter.includeFaults` parameter to true. For details, see *Advanced Settings* in the document *Administering webMethods Mediator*.

Displaying the Runtime Metrics

Use the following procedure to display runtime metrics for a virtualized asset.

Prerequisite:

You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see the section *Configuring Communication with CentraSite* in the document *Administering webMethods Mediator*.

To view the runtime metrics, the following prerequisites must be met:

- To view the runtime metrics of an API, it is necessary that the virtual type's definition includes the Runtime Metrics profile.
- If you do not see the Runtime Metrics profile of an API, it is probably because you do not have “View” permission for the profile.

▶ To display runtime metrics for an asset

1. In CentraSite Business UI, display the details page for the asset whose runtime metrics you want to view. If you need procedures for this step, see the section ***Viewing Details for an API Proxy***.
2. Open the **Runtime Metrics** profile.
3. Expand the **Filters** node.
4. Use the following fields to filter the metrics list you want to view:

In this field...	Specify...
Target	A target of the asset, or select **All** to view the metrics of all targets to which the virtual service is deployed. CentraSite displays **None** by default.
Date Range	A range of dates from which to view the metrics (e.g., Last 1 hour, Last 12 hours, Last 1 day, Last 5 days, Last 10 days, Last 20 days, Custom, etc.). CentraSite displays **Last 10 days** by default.
Start Date/End Date	If chosen **Custom** in the previous field, then the time period for which to view the metrics. **Start Date**: Click the calendar and select a starting date and time. **End Date**: Click the calendar and select a ending date and time.
Display Interval	A running count metrics of the service displayed at regular time intervals.

In this field...	Specify...
	The interval is specified in the format 3m 2d 6h; wherein m indicates the month, d indicates the day and h indicates the hour.

5 Click **Refine**.

CentraSite displays a graphical view of the metrics for all performance categories as shown below:

Multi-line Chart

The chart shows the Minimum Response Time, Maximum Response Time and Average Response Time of the API.

Pie Chart

The chart shows the Success Request Counts, Total Request Counts and Fault Counts of the API.

Gauge Chart

The chart shows the availability of the API.

The Runtime Events

CentraSite can receive the following predefined runtime event types.

To view the runtime metrics, the following prerequisites must be met:

- You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see the section *Configuring Communication with CentraSite* in the document *Administering webMethods Mediator*.
- You must configureCentraSite to receive run-time events from Mediator, as described in the section *Managing Targets and Run-Time Events* .

Event Type	Description
Lifecycle	A Lifecycle event occurs each time Mediator is started or shut down.
Error	An Error event occurs each time an invocation of an API results in an error.
Policy Violation	A Policy Violation event occurs each time an invocation of an API violates a run-time policy that was set for the API.
Transaction	A Transaction event occurs each time an API is invoked (successfully or unsuccessfully).

Event Type	Description
Monitoring	Mediator publishes key performance indicator (KPI) metrics, such as the average response time, fault count, and availability of all APIs (described below).

Notes:

1. For more information about runtime event types, see *Run-Time Event Notifications* in the document *Administering webMethods Mediator*.
2. For details about intervals, see *The Metrics Tracking Interval* in the document *Administering webMethods Mediator*.

Displaying the Runtime Events

Use the following procedure to display runtime events for a virtualized asset.

To view the runtime events, the following prerequisites must be met:

- To view the runtime events of an API, it is necessary that the virtual type's definition includes the **Runtime Events** profile.
- If you do not see the **Runtime Events** profile of an API, it is probably because you do not have “View” permission for the profile.

To display event information for an asset

1 In CentraSite Business UI, display the details page for the asset whose runtime events you want to view. If you need procedures for this step, see the section ***Viewing Details for an API Proxy***.

2 Open the **Runtime Events** profile.

3 Use the following fields to filter the event list you want to view:

In this field...	Specify...
Target	A target of the asset, or select **All** to view the event information of all targets to which the API is deployed. CentraSite displays **None** by default.
Consumer	A consumer of the asset, or select **All** to view the runtime event information of all consumers of the asset. CentraSite displays **All** by default. However, if you do not have at least one consumer registered in the registry, CentraSite displays **None** by default.
Event Type	A particular event type, or select **All** to view all event types. For a list of the supported event types, see the above section ***The Runtime Events***.

In this field...	Specify...
	CentraSite displays **All** by default.
Date Range	A range of dates from which to view the events (e.g., Last 1 hour, Last 12 hours, Last 1 day, Last 5 days, Last 10 days, Last 20 days, Last 1 month, Custom, etc.). CentraSite displays **Last 1 month** by default.
Start Date/End Date	If chosen **Custom** in the previous field, then the time period for which to view the metrics. **Start Date**: Click the calendar and select a starting date and time. **End Date**: Click the calendar and select a ending date and time.
Display Interval	A running count events of the service displayed at regular time intervals. The interval is specified in the format 3m 2d 6h; wherein m indicates the month, d indicates the day and h indicates the hour.

4 Click the **Refine** button.

5 Expand the **Graphical** node to display a graphical view of the event information.

6 Expand the **Tabular** node.

CentraSite displays a tabular view of the event information in the left pane.

Field	Description
Date/Time	The date/time that the event occurred. Click this hyperlinked value to view the **Event Detail** page, which will contain the event's SOAP request or response name in the Attribute column. Click the hyperlinked request or response name to display the full SOAP request or response.
Event Type	(Read-only.) The type of event (e.g., Monitoring, Policy Violation, Error, etc.).
Target	(Read only.) The target on which the event occurred.

7 To access the details of an event, click on the link for the event.

The **Event Details** dialog in the right pane shows a detailed information about the event that you select in the left pane.

12 Unpublishing an API Proxy

The following procedure describes how to unpublish a proxy API from one or more targets in webMethods Mediator.

To unpublish a proxy API from the targets, the following prerequisites must be met:

- If you are a registered user and accessing CentraSite using the logon credentials, to unpublish an API, you must at least have the instance-level Full permission on the API. If your user account belongs to a role that has the “Manage Assets” permission for an organization, you automatically have permission to unpublish all APIs in that organization. Also, if you are the owner of an API, you can unpublish that particular API.
- If you are a registered user and accessing CentraSite as a guest, to unpublish an API, you must at least have the instance-level Full permission on the API.
-

  The Unpublish action is *not visible*:

  - If you do not have a minimum of instance-level "Full" permission on the specified API.
  - If an API does not belong to one of the following asset types: Virtual Service, Virtual XML Service, Virtual REST Service.
  - If you do not have at least one target published to the webMethods Mediator.

You can unpublish a single proxy API or a selected set of proxy APIs. The descriptions in this section give you details on how to do this.

Unpublishing a Single API Proxy

You use this procedure to unpublish a single proxy API using it's **Unpublish** action.

To unpublish a proxy API

1. In CentraSite Business UI, display the details page of the proxy API you want to unpublish. For procedures, see the section *Viewing the Details of an API*.
2. On the API details page, click **Unpublish**. This opens the **Unpublish API** dialog.
3. From the **Available Targets** list, choose one or more targets from which you want to unpublish the proxy API.

   Or:

   If you want to unpublish the proxy API from all the targets, select the **All Targets** checkbox.

   When attempting to unpublish the proxy API, without selecting at least on target, a warning popup box appears
4. Click **Unpublish**.

   Or:

   If at any time you wish to terminate this operation, click **Cancel**.
5. An **Unpublish Progress** popup will display the progress state of unpublishing the proxy API from the webMethods Mediator.

   If the publish process failed, identify and correct the error and then try publishing the proxy API again.

Unpublishing a Set of Proxy APIs

You use this procedure to unpublish one or more proxy APIs using the native API's **Unpublish** action.

To unpublish multiple proxy APIs

1. In CentraSite Business UI, display the details page of the native API whose proxy endpoints (APIs) you want to unpublish from the webMethods Medaitor. For procedures, see the section *Viewing Details for an API*.
2. On the API details page, click **Unpublish**. This opens the **Unpublish API** dialog.
3. Choose the proxy APIs you want to unpublish from the drop-down labeled **Virtual Alias**. (The list will contain the proxy APIs that are defined for that particular API.)
4. From the **Available Targets** list, choose one or more targets from which you want to unpublish the proxy APIs.

   Or:

   If you want to unpublish the proxy APIs from all the targets, select the **All Targets** checkbox.

When attempting to unpublish the proxy APIs, without selecting at least on target, a warning popup box appears

5 Click **Unpublish**.

Or:

If at any time you wish to terminate this operation, click **Cancel**.

6 An **Unpublish Progress** popup will display the progress state of unpublishing the proxy APIs from the webMethods Mediator.

If the publish process failed, identify and correct the error and then try publishing the proxy APIs again.

13 Deleting an API Proxy

Deleting an API permanently removes the proxy API from the CentraSite registry.

To delete a proxy API, the following conditions must be met:

- You cannot delete the predefined APIs (not even if you have the default permissions associated with the CentraSite Administrator role).
- If you are not the owner of the API, you cannot delete the API unless you have “Manage Assets” permission (granted though a role-based permission) or at least Full permission on the API (granted through an instance-level permission).
- You cannot delete an API that is in pending state (e.g., awaiting approval).
- You cannot delete an API if any user in your CentraSite registry is currently modifying the API.
- Deleting an API will not remove the supporting documents that are attached to it.

You can delete a single proxy API or a selected set of proxy APIs. The descriptions in this section give you details on how to do this.

Deleting a Single API

Use the following procedure to delete a single API.

To delete an API

1. In CentraSite Business UI, display the details page for the API that you want to delete. If you need procedures for this step, see the section *Viewing the Details of an API*.
2. On the API’s actions menu, click **Delete** (🗑).
3. When you are prompted to confirm the delete operation, click **Yes**.

   The API is permanently removed from the CentraSite registry.

Deleting a Set of APIs

You can delete multiple APIs in a single step. The rules described above for deleting a single API apply also when deleting multiple APIs.

Use the following procedure to delete a set of APIs.

To delete multiple APIs in a single operation

1. In CentraSite Business UI, use either the *Browse* or *Search* feature to select a set of APIs you want to delete. If you need information on how to browse or search the CentraSite catalog, refer to the section *Browsing the CentraSite Catalog* or *Searching the CentraSite Catalog* in the document *Managing the CentraSite Catalog*.
2. Mark the checkbox next to the name of each API you want to delete.
3. In the actions menu, click **Delete** (🗑).

   Note: If one or more or the selected APIs is in pending state (e.g., awaiting approval), an error message will appear and no APIs will be deleted.

14 Privileged User of an API Proxy

A *Privileged User* is a user who has an elevated level of access to perform various actions on a proxy API, such as publishing the proxy API to webMethods Mediator or configuring the API key settings.

When the user tries to perform a privileged action on the proxy API, CentraSite validates the user's ID to determine whether the user holds the necessary privileges, and if so, it validates whether the privileges are enabled. If the user fails these validation, CentraSite does not perform the action.

A user who belongs to the "CentraSite Administrator" role defines the privileged user for the proxy API in the CentraSite configuration file (centrasite.xml).

```
<Service id="DeploymentService" privilegeUser="INTERNAL\Administrator" ↩
requiredRoles="API Publisher"> </Service>
```

wherein,

Input Parameter	Specifies...
`id`	A unique identifier of the proxy API.
`privilegeUser`	User who has privilege to access and execute various actions on the proxy API.
`requiredRoles`	A set of roles required to access and execute the various actions on the proxy API.

By default, an `INTERNAL\Administrator` with `API Publisher` role is configured as the privileged user for the proxy API.