

webMethods API-Portal Documentation Supplement

Version 9.7 Fix Pack

January 2015

This document applies to webMethods API-Portal Version 9.7 Fix Pack and to all subsequent releases.

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

Copyright © 2015 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

webMethods API-Portal 9.7 Feature Supplement

■ About This Supplement	3
■ Registering Users in API-Portal	3
■ Email Notification Templates and Tokens	8
■ Advanced Configuration	9
■ User Registration with Social Login	10
■ Purging User Registration Notifications	15
■ Connecting to Specific Servers	15

About This Supplement

This supplement documents the features introduced in webMethods API-Portal YAP_9.7_Fix2. Use this supplement in conjunction with the *webMethods API-Portal Administrator's Guide*. For complete information about the fix, see the readme file packaged with the fix.

Registering Users in API-Portal

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

- Register with API-Portal and provide an email address and create a password. Upon approval, API-Portal creates an account for the user in the UMC, and the user can log in with the email address and password provided at registration. Users can manage their own account details and change the password from the Profiles link in API-Portal.
- Use their credentials for a current social account (Google, Facebook). Upon approval, API-Portal notifies the user. At the first login, API-Portal creates an account for the user in the UMC. Users must use the social account to change account details and their password. If you want to allow requesters to use their social account, you need to configure it.

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

- **Approval workflows.** Ensure security while simplifying operations during user registration by using approval workflows within API-Portal for security credentials provisioning. With approval workflows, API-Portal can notify an administrator (or a group of approvers) about user registration requests and allow the approvers to approve or reject the requests. Upon approval, API-Portal notifies the requester by

email. If the user registered with an email address and password, API-Portal creates the user account at approval. If the user registered with a social account, API-Portal creates the user account when the user first logs in after approval.

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

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

By default, API-Portal stores all user registration approvals and email notifications. Depending on the volume of user registration activity for your portal, you may want to periodically purge the approval and email notification entries. For more information, see ["Purging User Registration Notifications" on page 15](#).

Before you configure the registration process, you need to:

- **Customize email notification templates.** Use the default text provided or customize the text as needed. You can use pre-defined tokens as placeholders for specific information. For more information, see ["Customizing Email Templates" on page 8](#).
- **Configure the SMTP mail server.** If you have not already done so, configure the SMTP server to enable API-Portal to send email notifications. For instructions, see the *webMethods API-Portal Administrator's Guide*.
- **Configure social accounts.** If you want to allow requesters to use a social account to access the portal, you need to configure that access. For instructions, see ["User Registration with Social Login" on page 10](#).

Configuring Approval Workflows for User Registration

When a requester clicks **Register** on the API-Portal landing page or clicks **Log In** and enters his social login credentials, a registration request appears in the approver's Pending Approvals page in API-Portal. If a requester submits another request while the first request is still pending, API-Portal automatically rejects the second request.

By default, users with the API Administrator role can approve or reject user registration requests. To have additional users who can participate in the approval process, add the users to the approver group. API-Portal provides a pre-defined approver group, **API User Registration Approvers**.

To use approval workflows you need to:

1. **Assign users as approvers by adding them to the approver group.** Determine who in your organization will review user registration requests, and approve or reject them. In the UMC, add users who are to review and approve or reject registration requests to the approver group.
2. **Define the workflow approval process.** In API-Portal, specify that you want to require approval for all registration requests, and select at which points in the approval process API-Portal is to send email notifications.

API-Portal will automatically approve registration in the following situations:


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

Assigning Users to the Approver Group

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

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

To add or remove users as approvers

1. Start a web browser and go to the API-Portal UMC application at: `http://host:port/umc`
2. Click **User Management**, and then click **User groups**.
3. Click the **API User Registration Approvers** user group name, and then click **Associated Users**.
4. Click  **Edit Assignment**. Add or remove users as approvers as follows:
 - **To add users:** Select the users you want to add from the **Available items** box, and then click **Add**. The selected users are transferred to the **Assigned items** box, and can now receive and approve user registration requests.
 - **To remove users:** Select the users you want to remove from the **Assigned items** box, and then click **Remove**. The selected users are moved to the **Available items** box, and can no longer receive registration requests.
 - **To add or remove all users at once:** Click **Add All** or **Remove All**.
5. Click **Save**.

Configure Your Approval Workflow

To configure your workflow approval, you need to specify at which approval steps to send an email notification, and what the email subject and content is.

To configure the workflow approval

1. In API-Portal, click **Administration > Define registration process**.
2. Click **Require Approval**.
3. Select the notifications that you want to send for each workflow step that you want to use. For each step, the **Subject** and **Content** fields contain the content that will be used for all notifications sent from API-Portal. Use the default content or change the content, if needed. For more information, see "[Email Notification Templates and Tokens](#)" on page 8.
4. Click **Apply**.

Working with Pending Approvals

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

For additional information about the Pending Approvals page, see the *webMethods API-Portal Online Help*.

To approve or reject pending approvals

1. In API-Portal, select **Pending Approvals** from your user menu.
2. On the Pending Approvals page, review all the pending requests.
 - **To approve a request:** Select the request and then click **Approve**. At the confirmation dialog, click **OK**.
 - **To reject a request:** Select the request and then click **Reject**. At the confirmation dialog, provide a reason for the rejection, and then click **Reject**. The reason text will be included in the email notification sent to the requester.
 - **To approve or reject multiple requests:** Select all requests and then click **Approve** or **Reject**.

Configuring Email Confirmation for User Registration

Upon receiving a registration request, API-Portal can send an email notification to the requester at the email address provided at registration or to the social account email address. The email contains an activation link that the requester clicks to access the portal. Email confirmation is the default.

To configure email confirmation

1. In API-Portal, select **Administration > Define registration process**.
2. Click **Require Email Confirmation**.
3. The **Subject** and **Content** fields contain the content that will be used for all notifications sent from API-Portal. Use the default content or change the content, if needed. For more information, see "[Email Notification Templates and Tokens](#)" on page 8.
4. Click **Apply**.

Configuring Automatic User Registration

With automatic registration, API-Portal automatically processes and approves all registration requests and social logins upon receipt. API-Portal notifies the user by email that their account is activated and ready to use. The user needs to log in to the portal using the email address or social account credentials they registered with.

To automatically process user registrations

1. In API-Portal, select **Administration > Automatic Registration**.
2. The **Subject** and **Content** fields contain the content that will be used for all notifications sent from API-Portal. Use the default content or change the content, if needed. For more information, see "[Email Notification Templates and Tokens](#)" on page 8.
3. Click **Apply**.

Viewing and Managing Users

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

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

Note: Social account passwords cannot be changed through the UMC.

From the UMC, an administrator can view and change the user details, such as the email address and phone number, groups the user is assigned to, and privileges assigned to the user. You can also delete a user and its associated account. For more information about user management, see the *webMethods API-Portal Online Help*.

Email Notification Templates and Tokens

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

Customizing Email Templates

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

To change the content of email templates

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

Using Email Tokens

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

Token	Will be replaced with...
@approver.name	The name of the person who approved the request.
@requester.name	The name of the user who submitted the registration request.
@comment	When used in an approval workflow, this token will be replaced with the reason the approver entered when rejecting the request.
@api-portal.url	A link to the URL of the API-Portal that the user is registered to use.
@activation.token	The access token the user needs to access API-Portal. This token is valid only in email confirmation.

Advanced Configuration

Some user registration parameters are configurable by a REST service. To configure user registration parameters using a REST service, you must have API Administrator privileges. Any REST client tool can be used to invoke the mentioned APIs.

REST Resource	Parameters	Description
Active user approver group	Endpoint: http://host:port/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.approvergroup/ Method: POST Request Body: <i>New approver group name</i>	API User Registration Approvers. Sample Request: Endpoint: http://mcsv01.eur.ad.sag/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.approvergroup/ Method: POST Request Body: Approvers
Onboarding request token	Endpoint: http://host:port/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.validateemail.token.ttl/ Method: POST Request Body: <i>New expiration period in minutes</i>	REST Service to change the expiration period of the link generated during email confirmation workflow. The default value is 30 minutes. Sample Request: Endpoint: http://mcsv01.eur.ad.sag/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.validateemail.token.ttl/ Method: POST Request Body: 60
Data purging	Endpoint: http://host:port/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.purge/ Method: POST Request Body: <code>true false</code>	REST Service to modify the default purging behavior for the approval objects. The default value is true . If the value is set to false, the API Administrator must perform manual purging.

REST Resource	Parameters	Description
		<p>Metadata for pending approvals will be deleted after the requester is approved or rejected.</p> <p>By this time, the token, which we see in the URL, will also be deleted.</p> <p>API-Portal cannot differentiate between an invalid token (any token entered by the requester) and a valid, but expired, token (which is deleted by the first request).</p> <p>Sample Request:</p> <p>Endpoint: http://mcsv01.eur.ad.sag/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.purge/</p> <p>Method: POST</p> <p>Request Body: false</p>

The expected HTTP response code for all REST resources listed in the table is 202.

User Registration with Social Login

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

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

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

When you allow social login to API-Portal:

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

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

What is OAuth?

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

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

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

Configuring Google Login

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

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

To use Google for user registration and log in

1. Log in to developers.google.com and create an app for API-Portal registration.
 - a. Go to the Google Developer Documentation.
 - b. Log in using your Google account credentials.
 - c. Click **Create Project**. Provide the project name and project id, and then click **Create**.
 - d. Click on the link for your project. You will be taken to your project. In the left side navigation, go to **APIs & auth > Credentials**, and then click **Create new Client ID**.
 - e. Select the application type **Web application**. The warning says that a product name needs to be set. Click **Configure consent screen** and set the product name.
 - f. At the Create Client ID screen, the application requires that the domain be public. Provide the URL and the relevant port of your website in the **Authorized Javascript Origins** field, and provide the redirect URL and the relevant port in the **Authorized Redirect URIs** field. The URLs must be HTTPS. Click **Create Client ID**.
 - g. In the left side navigation, go to **APIs & auth > Credentials**. You will find the client id and client secret on the right.
 - h. Make note of the client key and the client secret from the Google app, because you will need it in the next step.
 - i. In the left side navigation, go to **APIs & auth > APIs**. Search for Google+ API and enable it. You will then find the Google+ API in the **Enabled APIs** section.
2. Log in to UMC as Administrator, and configure the OAuth settings for the Google app as described in "[OAuth Properties for Social Login](#)" on page 13.
3. If you have not already done so, choose the level of approval needed, as described in "[Registering Users in API-Portal](#)" on page 3.

Configuring Facebook Login

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

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

To use Facebook for user registration and log in

1. Log in to developers.facebook.com and create an app for API-Portal registration.
 - a. Go to: <http://developers.facebook.com>
 - b. Log in using your Facebook account credentials.
 - c. Go to **Apps > Add a New App**.
 - d. On the Add a New App page, click **Website**.
 - e. At the next screen, enter a name for your application and click **Create New Facebook App ID**.
 - f. Choose the category **Apps For Pages**.
 - g. Scroll down. In the **Tell us about your website** section, provide the URL of your website. The website URL should be an HTTPS URL and include the corresponding port.
 - h. Click **Skip Quick Start**. This will create your Facebook app and display the Dashboard for your app.
 - i. Go to the Settings page by clicking the **Settings** link in the left side navigation. Provide a valid contact email here. This is required to make your application public to all users.
 - j. Go to the Status & Review page by clicking the **Status & Review** link in the left side navigation.
 - k. Make your application public by toggling the button from **No** to **Yes**. If the application is not made public, only the developer and the test users of the application will be allowed to log in using the app.
 - l. Make note of the client key (**App ID**) and the **App Secret**, because you will need it in the next step.
2. Log in to UMC as Administrator, and configure the OAuth settings for the Facebook app, as described in "[OAuth Properties for Social Login](#)" on page 13.
3. If you have not already done so, choose the level of approval needed, as described in "[Registering Users in API-Portal](#)" on page 3.

OAuth Properties for Social Login

If you are using social logins to provide access to your API-Portal, you need to configure OAuth settings in UMC to authorize the portal to work with the social apps. Log in to UMC as Administrator, and then click the Configuration tab. Select **OAuth** from the drop-down box to show only the OAuth properties. Set the properties as described below.

com.aris.umc.oauth

com.aris.umc.oauth.active

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

com.aris.umc.oauth.api.keys

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

com.aris.umc.oauth.api.secrets

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

com.aris.umc.oauth.debug

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

com.aris.umc.oauth.providers

A comma-separated list of OAuth providers for each social app used for login. Both values are optional. The list of providers specified here defines how many login possibilities are displayed. If, e.g. only Google is configured, then the login page will show **Login with Google** only.

com.aris.umc.oauth.tenant

Specifies the default tenant used for social authentication. This value is read-only.

Sample Configuration for Google

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

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

Removing Social Login

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

To remove social registration and login access

1. Log in to UMC as Administrator, and disable OAuth:
 - a. Go to the API-Portal UMC application at: `http://host:port/umc`, and log in with your administrator credentials.
 - b. Click the Configuration tab.
 - c. Set the `com.aris.umc.oauth.active` property to `false`.
2. Log out from the UMC.

Purging User Registration Notifications

By default, API-Portal stores all user registration approvals and email notifications. Depending on the volume of user registration activity for your portal, you may want to periodically purge the approval and email notification entries. To purge registration notifications, execute the Purge REST service with the following properties:

Endpoint: `http://host:port/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.purge/`

Method: POST

Request Body:

`true|false`

The expected HTTP response code is 202.

Sample Request

Endpoint: `http://mcsv01.eur.ad.sag/abs/apirepository/configurations/com.aris.umc.apiportal.useronboarding.approval.purge/`

Method: POST

Request Body: `false`

The expected HTTP response code is 202.

Connecting to Specific Servers

If you need to connect to specific servers instead of going through a proxy server, set the `apiportal.proxy.noProxyHosts` parameter to list those servers.

You can find this parameter in the `apiportal.properties` file. The file is located in the `Software AG_directory\API-Portal\server\bin\work\work_abs_m\base\webapps\abs\WEB-INF\classes\com\aris\modeling\server\webapp\apiportal\configuration` directory.

apiportal.proxy.noProxyHosts

apiportal.proxy.noProxyHosts

Specifies the servers that API-Portal connects to directly and not through a proxy server.

Use the format *host_1* | *host_2* | *host_n*

where *host* is the host name or IP address of the server. You can specify a list of servers, each separated by a |. You can also specify a wildcard character (*) for matching.

For example:

```
apiportal.proxy.noProxyHosts=localhost|127.0.0.1|*.int.abc.org
```