WSO2 API Manager 3.2.0 Developer Fundamentals

Working with Published APIs

WSO2 Training

Publisher APIs

WSO2 API MANAGER PUBLISHER

API

- Create/Update/Delete/Retrieve API / API Documentation
- Import/Retrieve/Update API definition
- Retrieve subscription throttling policies of API
- Create new API version
- Get/Update resource policies
- Life cycle state changes
- Validate API attribute/definition/endpoint
- Retrieve/Add schema to a GraphQL API
- Monetize API /Get status/details of monetized API
- Add/Retrieve/Delete/Update mediation policies
- Upload/Update/Retrieve/Download client certs
- Generate mock response payloads
- Get/Upload thumbnail image
- Retrieve the ARNs of AWS Lambda functions
- Get security Audit Report
- Publish to External Stores / Get external store lists

Subscriptions

- Retrieve/Block/Unblock subscription
- Get details of subscribed user

Throttling Policies

- Retrieve throttling policies

Endpoint Certificates

- Upload/Update/Retrieve/Download endpoint certificates

Alerts

- Add/Delete/Retrieve alerts subscriptions
- Retrieve/Update/Delete alert configuration

API Product

- Create/Update/Retrieve/Delete API product
- Add/Update/Retrieve/Delete API product Documentation
- Get / Upload thumbnail

Other

- Validate roles
- Get/Add/Update/Delete/Validate shared scopes
- Download mediation policies
- Retrieve settings
- Retrieve gateway environments
- Retrieve tenants
- Retrieve all registered labels
- Retrieve alert details and configurations
- Get API categories
- Get all Key Managers

Link - Publisher APIs

Developer Portal APIs

API

- Retrieve API
- Retrieve API definition
- Retrieve subscription throttling policies of API
- Generate SDK
- Retrieve supported SDK list
- Retrieve API documentation
- Add/Update/Retrieve/Delete API ratings
- Add/Update/Retrieve/Delete comments
- Get thumbnail image

Throttling Policies

- Retrieve throttling policies

Applications

- Add/Update/Retrieve/Delete application
- Generate/Retrieve/Map/Re-generate/Cleanup application keys
- Update grant type and callback urls
- Regenerate consumer secrets
- Retrieve details of keys types
- Generate application token
- Generate/Revoke API key
- Import/Export applications

Alerts

- Add/Delete/Retrieve alerts subscriptions
- Retrieve/Update/Delete alert configuration

Subscriptions

- Retrieve/Add/Remove/Update subscription
- Get invoice for monetized API

Other

- Retrieve all tags
- Register new user(self-signup)
- Retrieve settings
- Retrieve application attributes
- Retrieve tenants by state
- Give API Recommendations
- Get API categories
- Get all Key Managers
- Retrieve GraphQL policies
- Change user password

Link - Developer Portal APIs

Admin APIs

WSO2 APIM ADMIN PORTAL

API and API Product

- Import/Export API
- Import/Export API Product
- Retrieve API categories
- Add/Update/Delete API Category

Alerts

- Retrieve/Subscribe/Unsubscribe alerts
- Retrieve/Subscribe/Unsubscribe bot detection alerts
- Get all bot detected data

Custom Rules

- Retrieve custom rules
- Add/Delete/Update custom rules

Applications

- Retrieve/Remove Applications
- Retrieve Application policies
- Add/Delete/Update Application policy
- Change Application Owner
- Export/Import Application

Policies

- Add/Delete/Update subscription/mediation policy
- Retrieve subscription/medation policies

Scopes

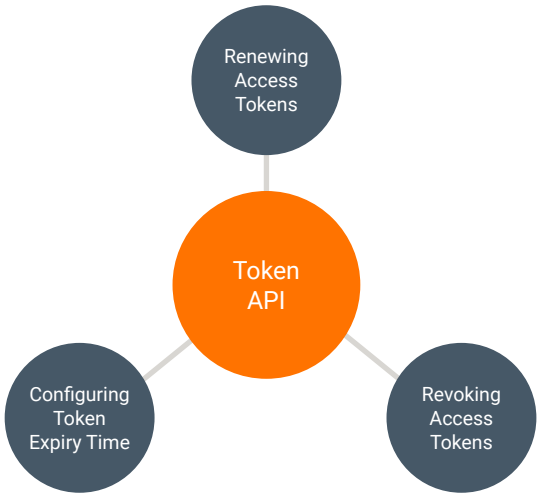
- Retrieve scopes
- Add/Update roles
- Retrieve Role/scope mappings

Other

- Retrieve advanced throttling policies
- Add/Update/Delete advanced throttling policy
- Retrieve blocking conditions
- Add/Update/Delete blocking conditions
- Retrieve/Add/Update/Delete labels
- Publish usage records
- Get the monetization status publisher usage
- Retrieve pending workflow details
- Update workflow status
- Get tenant details
- Retrieve settings
- Add/Update/Delete/Retrieve Key Manager
- Import/Export tenant theme

Link - Admin APIs

Token API



Link - Token API

Users need access tokens to invoke APIs subscribed under an application. Access tokens are passed in the HTTP header when invoking APIs. The API Manager provides a Token API that you can use to generate and renew user and application access tokens. The response of the Token API is a JSON message. You extract the token from the JSON and pass it with an HTTP Authorization header to access the API.

WSO2 API Manager supports the four most common authorization grant types and you can also define additional types.

- SAML Extension Grant Type
- Authorization Code Grant Type
- NTLM Grant Type
- Password Grant Type

Renewing Access Tokens

After an access token is generated, sometimes you might have to renew the old token due to expiration or security concerns. You can renew an access token using a refresh token, by issuing a REST call to the Token API with the following parameters.

- Payload
- Headers

Ex: - curl -k -d "grant_type=refresh_token&refresh_token=<refreshtoken>&scope=PRODUCTION" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token

The Token API URL is https://localhost:8243/token, assuming that both the client and the Gateway are run on the same server.

- payload: "grant_type=refresh_token&refresh_token=<refreshtoken>&scope=PRODUCTION". Replace the <retoken> value with the refresh token generated in the previous section.
- headers: Authorization :Basic <base64 (client id : client secret)>, Content-Type: application/x-www-form-urlencoded.

For example, the following cURL command can be used to access the Token API.
curl -k -d "grant_type=refresh_token&refresh_token=<refreshtoken>&scope=PRODUCTION" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token

The above REST message grants you a renewed access token along with a refresh token, which you can use the next time you renew the access token. A refresh token can be used only once. You can configure an expiration time for the refresh token by adding following configuration in the <APIM_HOME>/repository/conf/deployment.toml file.

```
[oauth.token_validation]

 refresh_token_validity = 3600
```

Revoking Access Tokens

After issuing an access token, a user or an admin can revoke it in case of theft or a security violation. You can do this by calling Revoke API using a utility like cURL.

The Revoke API's endpoint URL is : *https://localhost:8243/revoke*.
Parameters required to invoke this API are as follows:

- The token to be revoked
- Consumer key and consumer secret key. Must be encoded using Base64 algorithm
- Token type hint - this optional parameter specifies whether the token to be revoked is access token or refresh token

Ex: - `curl -k -d "token=<ACCESS_TOKEN_TO_BE_REVOKED>" -H "Authorization: Basic Base64Encoded(Consumer key:consumer secret)" http://localhost:8280/revoke.`

For example, `curl -k -d "token=<ACCESS_TOKEN_TO_BE_REVOKED>" -H "Authorization: Basic Base64Encoded(Consumer key:consumer secret)" http://localhost:8280/revoke.`

Configuring the Token Expiration Time

User access tokens have a fixed expiration time, which is set to 60 minutes by default. Before deploying the API Manager to users, extend the default expiration time by adding or updating the token_validation.app_access_token_validity value under the [oauth] section in *deployment.toml* file.

```
[oauth]
token_validation.app_access_token_validity = 10000
```

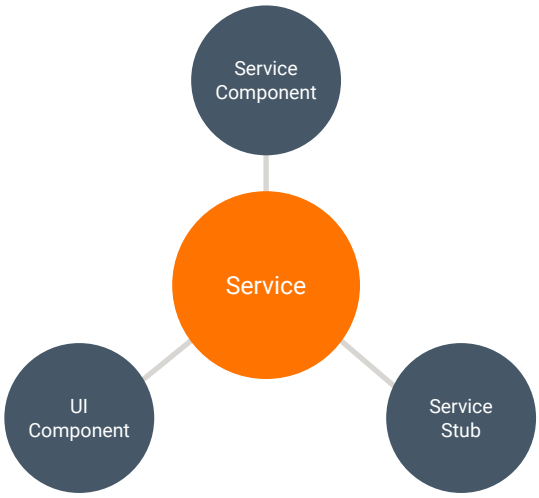

Also take the timestamp skew into account when configuring the expiration time. The timestamp skew is used to manage small time gaps in the system clocks of different servers.

For example, let's say you have two Key Managers and you generate a token from the first one and authenticate with the other. If the second server's clock runs 300 seconds ahead, you can configure a 300s timestamp skew in the first server. When the first Key Manager generates a token (e.g., with the default life span, which is 3600 seconds), the timestamp skew is deducted from the token's life span. The new life span is 3300 seconds and the first server calls the second server after 3200 seconds.

You configure the timestamp skew using in deployment.toml in <PRODUCT_HOME>/repository/conf

```
[oauth]
timestamp_skew = 100
```

Admin Services



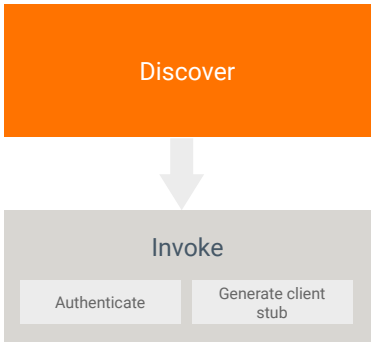
Link - Admin Services

WSO2 products are managed internally using SOAP Web services known as **admin services**. WSO2 products come with a management console UI, which communicates with these admin services to facilitate administration capabilities through the UI.

A service in WSO2 products is defined by the following components:

- Service component: provides the actual service
- UI component: provides the Web user interface to the service
- Service stub: provides the interface to invoke the service generated from the service WSDL

Admin Services



There can be instances where you want to call back-end Web services directly. For example, in test automation, to minimize the overhead of having to change automation scripts whenever a UI change happens, developers prefer to call the underlying services in scripts. In order to do this, you need to discover and invoke these services from your applications.

By default, the WSDLs of admin services are hidden from consumers. Given below is how to discover them.

- Add the following configuration to the <PRODUCT_HOME>/repository/conf/deployment.toml file.
  [admin_service.wsdl]
  enable = true
- Restart the server.
- Start the WSO2 product with the -DosgiConsole option, such as sh <PRODUCT_HOME>/bin/wso2server.sh -DosgiConsole in Linux.
- When the server is started, hit the enter/return key several times to get the OSGI shell in the console.
- In the OSGI shell, type: osgi> listAdminServices
- The list of admin services of your product are listed.
- To see the service contract of an admin service, select the admin service's URL and then paste it in your browser with ?wsdl at the end.
  For example : https://localhost:9443/services/UserAdmin?wsdl
- In products like WSO2 ESB and WSO2 API Manager, the port is 8243 (assuming 0 port offset). However, you should be accessing the Admin Services via the management console port, which is 9443 when there is no

- port offset.
- Note that the admin service's URL appears as follows in the list you discovered in step 6: AuthenticationAdmin, AuthenticationAdmin, https://<host IP>:8243/services/AuthenticationAdmin

Let’s try it out!

Using Published APIs