



# TIBCO® Order Management

## Security Guidelines

Version 6.1.0 | October 2024

# Contents

---

<b>Contents</b>	<b>2</b>
<b>About this Product</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Security Features</b>	<b>5</b>
<b>Security Vulnerabilities</b>	<b>6</b>
<b>Ensuring TIBCO Order Management Security</b>	<b>7</b>
Authorization Service	7
Registering a Tenant	8
Update tenant information	12
Get tenant information	13
Delete tenant	13
Create User	14
Update User	15
Get User	16
Delete User	17
Generating an authorization token	18
Configuring SSL for TIBCO Order Management	22
Encrypt Password Utility	24
<b>TIBCO Documentation and Support Services</b>	<b>25</b>
<b>Legal and Third-Party Notices</b>	<b>27</b>

# About this Product

---

TIBCO® Order Management is an elastic, catalog-driven order management system for digital service providers. It accepts orders from any customer engagement system and orchestrates the tasks required for fulfilling the orders.

TIBCO Order Management is the next generation of TIBCO® Fulfillment Order Management and partially replaces the old product. To better align TIBCO Fulfillment Order Management with market demand, the product's capabilities have been reorganized into two new products: TIBCO® Order Management and TIBCO® Offer and Price Engine.

# Introduction

---

This document describes guidelines to ensure security within the various components of TIBCO® Order Management. It also provides additional security-related guidance and recommendations for other aspects of internal and external communication. In particular, this document provides details of product connectivity and configuration of security options.

# Security Features

---

TIBCO® Order Management includes the following security features.

- Secure transports for communications among application peer processes.
- HTTPS for secure connections.
- Authentication and authorization service.

# Security Vulnerabilities

---

This topic describes the key security technologies for TIBCO Order Management. In addition to these key technologies, security also depends in part on the correct configuration and use of its components and capabilities.

## **OpenSSL**

Security features that protect TIBCO Order Management connections and communications depend on the implementation of OpenSSL. If the security of OpenSSL were compromised, TIBCO Order Management and applications that use TIBCO Order Management could be vulnerable as well.

# Ensuring TIBCO Order Management Security

---

To ensure security within and among the components of TIBCO Order Management, the following security provisions are provided.

- [Authorization Service](#)
- [Configuring SSL for TIBCO Order Management](#)
- [Encrypt Password Utility](#)

## Authorization Service

To ensure secure access to TIBCO Order Management system REST APIs and support multitenancy, token-based authentication is implemented in TIBCO Order Management.

The authentication service in TIBCO Order Management uses the JSON Web Token (JWT) to validate user credentials (user name, password, and tenantID).

The following functions are covered under the Authorization Service:

- [Registering a Tenant](#)
- [Update tenant information](#)
- [Get tenant information](#)
- [Delete tenant](#)
- [Create User](#)
- [Update User](#)
- [Get User](#)
- [Delete User](#)

After a user is created, authenticate it by following the procedures in [Generating an authorization token](#) topic.

## Registering a Tenant

You can register a tenant by setting the identity provider type to Oracle, PostgreSQL, LDAP, or EXTERNAL. Separate databases are created for each registered tenant's user.

Tenant registration API is shown as follows:

This operation registers tenant information. This API can handle only single tenant registration at a time.

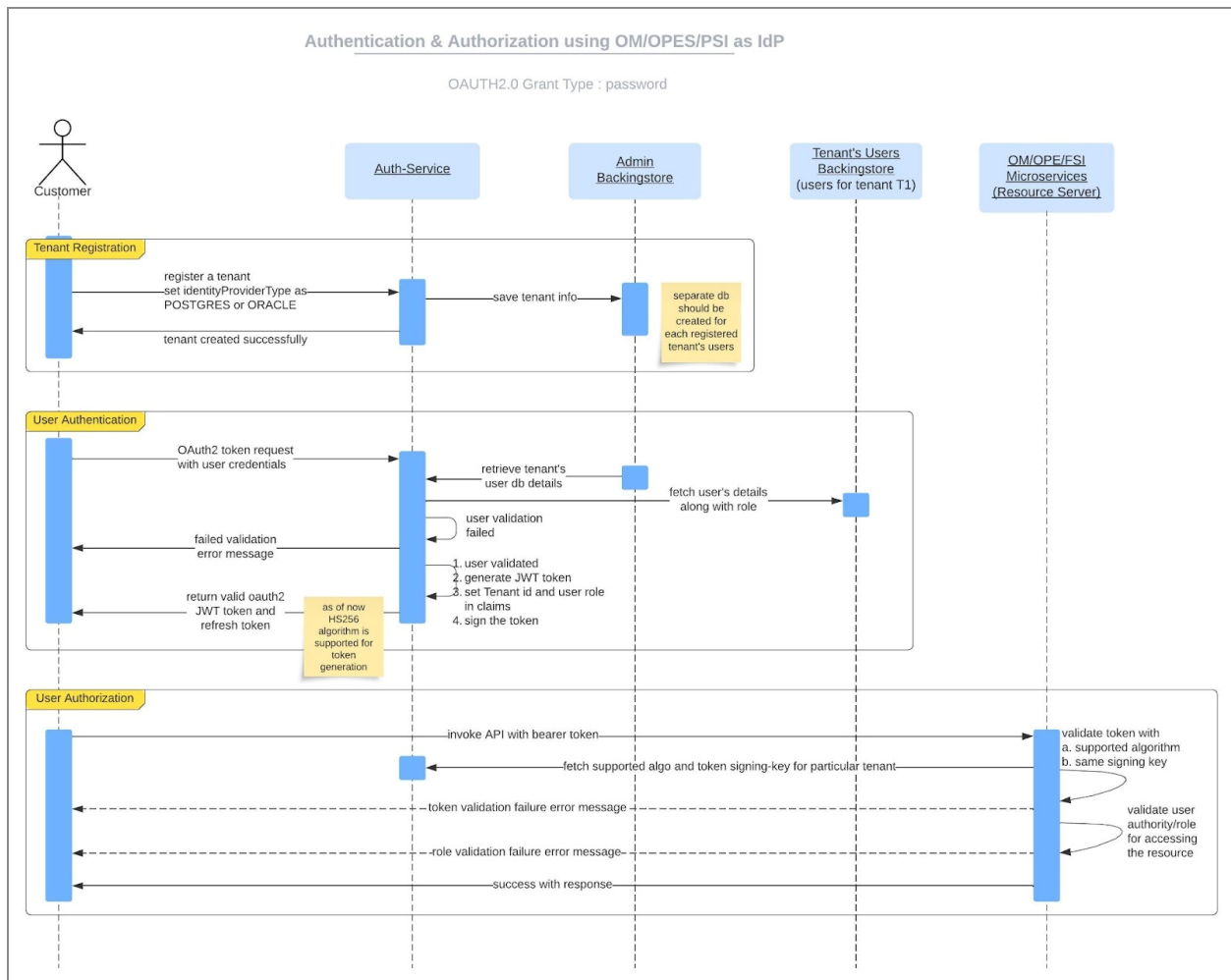
**Method:** HTTP POST

**Endpoint:** `http://<host_address>:<port_address>/v1/tenant`

Parameter	Cardinality	Description
X-API-AppId	Mandatory	The application ID is used for getting the user details.
X-API-Key	Mandatory	This key is used for getting the user details.

If you set the identity provider as Oracle or PostgreSQL, then you have to create separate databases for each tenant.





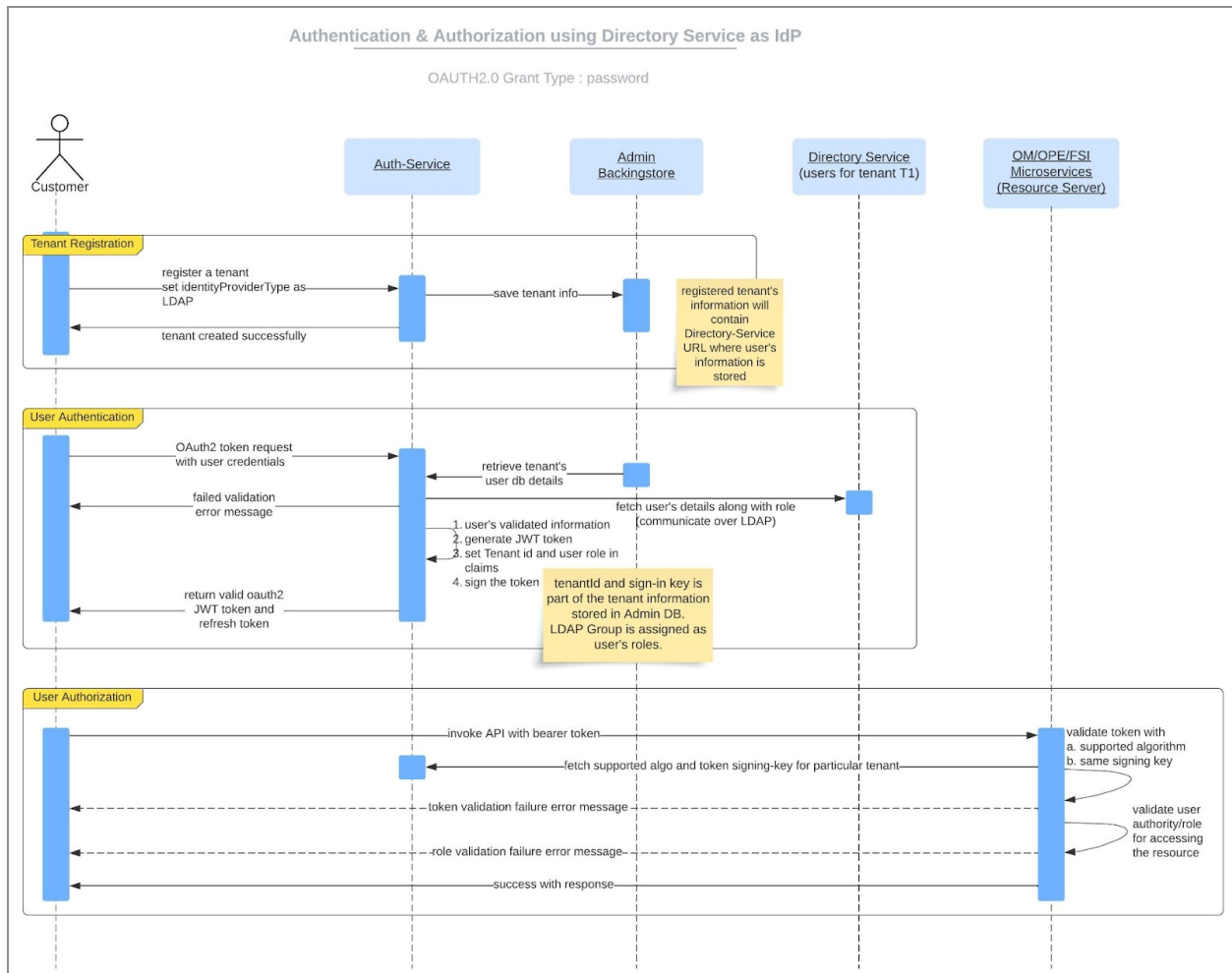
The following sample is shown for RelationalSchema (Postgres/Oracle) identityProviderType:

```

{
  "tenantId": "TIBCO",
  "clientId": "order",
  "clientSecret": "order",
  "identityProviderType": "POSTGRES",
  "supportAlgorithm": "HS256",
  "signingKey": "100f4c1f-f333-4c25-bd8c-e4809722b6a7",
  "relationalSchema": {
    "dataSourceURL":
    "jdbc:postgresql://localhost:5432/userdb11?currentSchema=userschema11",
    "dataSourceUserName": "user11",
    "dataSourcePassword": "user11"
  }
}

```

When you have set the identity provider as LDAP, all the users and their roles are maintained in some Directory service.



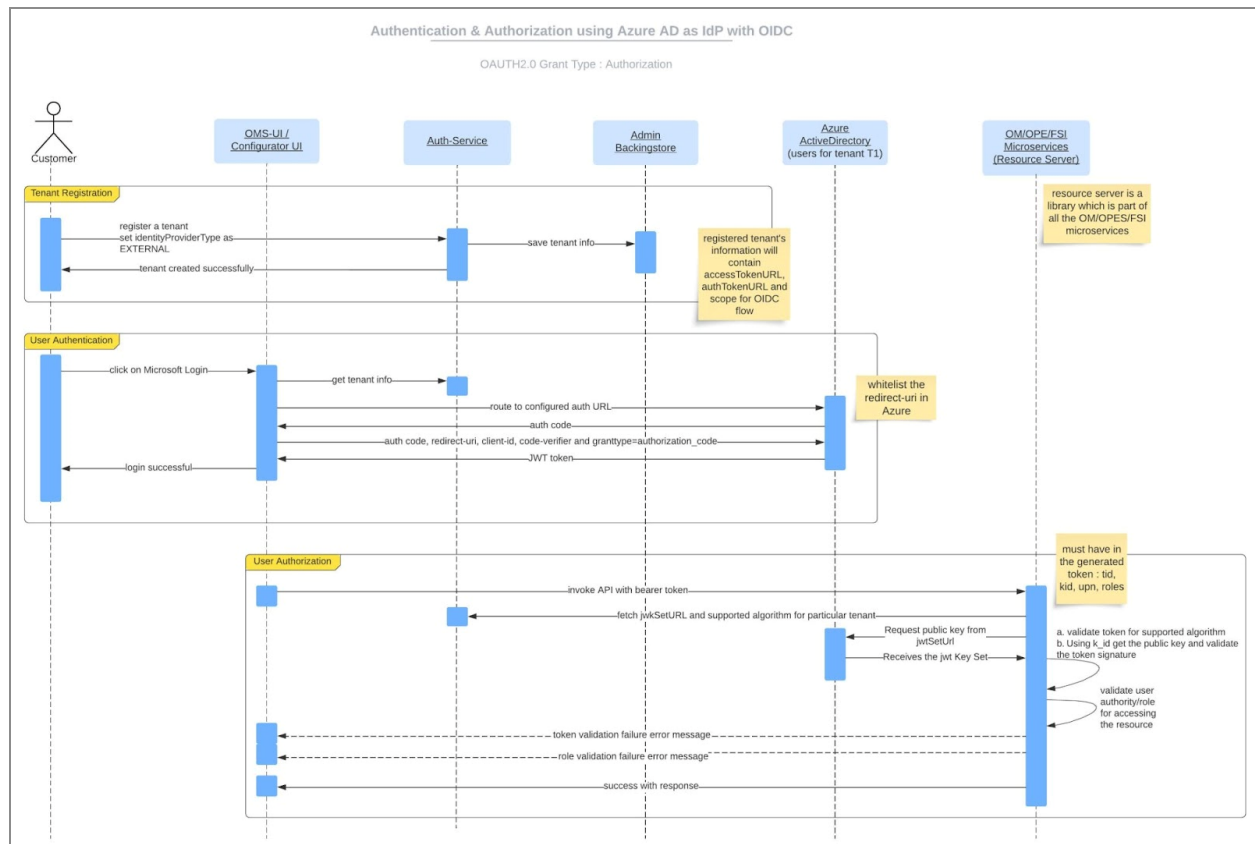
The following sample is shown for LDAP identityProviderType:

```

{
  "tenantId": "TIBCOLDAP",
  "clientId": "tibco-ldap-client",
  "clientSecret": "tibco-ldap-secret",
  "identityProviderType": "LDAP",
  "supportAlgorithm": "HS256",
  "signingKey": "100f4c1f-f333-4c25-bd8c-e4809722b6a7",
  "ldapSchema": {
    "ldapURLForDirectoryService": "string",
    "directoryServiceDomainName": "string",
    "directoryServiceRootDistinguishedName": "string"
  }
}

```

When you have set identity provider as EXTERNAL, you do not have to use the Order Management's Authentication service for user authentication and token generation. As of now, we support Microsoft Azure Active-Directory as the external authentication service. Even when you have set the identity provider as EXTERNAL, the tenant information is still stored in the Order Management's Authentication service's relational database.



The following sample is shown for EXTERNAL identityProviderType:

```

{
  "tenantId": "string",
  "clientId": "string",
  "clientSecret": "string",
  "identityProviderType": "EXTERNAL",
  "signingKey": "100f4c1f-f333-4c25-bd8c-e4809722b6a7",
  "supportAlgorithm": "RS256",
  "jwkSetUrl": "string",
  "issuer": "string",
  "oidcSchema": {
    "authUrl": "string",
    "accessTokenUrl": "string",
    "scope": "string"
  }
}
  
```

```
}
}
```

Authorization service can generate a token for all tenants. Each tenant can have a different token algorithm. The following algorithms are supported:

- HMAC (HS256, HS384, HS512)
- RSA (RS256, RS384, RS512)

Order Management Authorization service generates token with HS256. All services can decode or handle any of the above algorithms.



**Note:**

- Supported algorithms must match with one, which is used at the time of registration, This is used for validating tokens (Only in the case of RSA).
- Issuer is validated during registration while validating the token.

## Update tenant information

This operation updates tenant information if the tenant details are already present in database.

**Method:** HTTP PUT

**Endpoint:** `http://<host_address>:<port_address>/v1/tenant`

### Get User Parameters

Parameter	Cardinality	Description
X-API-AppId	Mandatory	The application ID is used for getting the user details.
X-API-Key	Mandatory	This key is used for getting the user details.

For more information on various identityProviderType scenarios, see the sample from the 'Register tenant' topic in *TIBCO® Order Management Web Services Guide*.

## Get tenant information

This operation is used to get the tenant information if it is already present in database.

**Method:** HTTP GET

**Endpoint:** http://<host\_address>:<port\_address>/v1/tenant

*Get User Parameters*

Parameter	Cardinality	Description
tenantId	Mandatory	This is the TENANT value as stored in the users table in the database.
X-API-AppId	Mandatory	The application ID is used for getting the user details.
X-API-Key	Mandatory	This key is used for getting the user details.

## Delete tenant

This operation deletes tenant information if the tenant is already present in database.

**Method:** HTTP DELETE

**Endpoint:** http://<host\_address>:<port\_address>/v1/tenant

*Get User Parameters*

Parameter	Cardinality	Description
tenantId	Mandatory	This is the TENANT value as stored in the users table in the database.
X-API-AppId	Mandatory	The application ID is used for getting the user details.
X-API-Key	Mandatory	This key is used for getting the user details.

## Create User

This request is used to create users.

**Method:** HTTP POST method

**Endpoint:** `http://<host_address>:<port_address>/v1/user`

### Create User Parameters

Parameter		Cardinality	Description
tenantId		Mandatory	This is the TENANT value as stored in the users table in the database. If the tenantId is not present in the database, then a new TENANT is created.
X-API-AppId		Mandatory	The application ID is used for getting user details. The default ID is auth.
X-API-Key		Mandatory	This key is used for getting user details. The default ID is auth.
userInfo (Body)	enabled	Mandatory	The value can be true or false. true makes the user accessible through the configurator and Order Management System UI and false makes the user disable.
	password	Mandatory	The password to be used for the user.
	Username	Mandatory	It specifies the user name to be created or modified.
	userRoles	Mandatory	It assigns the role to the user.  The default valid role values are ROLE_ADMIN and ROLE_USER. You can override the default roles if required.



**Note:** If the userName and tenantId provided in the request exist, then the user is modified with the provided values.

Example for the Create User request:

```
{
  "user": [
    {
      "password": "string",
      "userName": "string",
      "enabled": true,
      "userRoles": [
        "string"
      ]
    }
  ]
}
```

## Update User

This request is used to update an existing user.

**Method:** HTTP PUT method

**Endpoint:** `http://<host_address>:<port_address>/v1/user`

### *Update User Parameters*

Parameter		Cardinality	Description
tenantId		Mandatory	This is the TENANT value as stored in the users table in the database. If the tenantId is not present in the database, then a new TENANT is created.
X-API-AppId		Mandatory	The application ID is used for getting user details. The default ID is auth.
X-API-Key		Mandatory	This key is used for getting user details. The default ID is auth.
userInfo (Body)	enabled	Mandatory	The value can be true or false. true makes the user accessible through the configurator and Order Management System UI and false makes the user disable.

Parameter	Cardinality	Description
password	Mandatory	The password to be used for the user.
userName	Mandatory	It specifies the user name to be created or modified.
userRoles	Mandatory	It assigns the role to the user. The valid role values are ROLE_ADMIN and ROLE_USER.

**Note:** If the userName and tenantId provided in the request exist, then the user is modified with the provided values.

Example for the Update User request:

```
{
  "user": [
    {
      "password": "string",
      "userName": "string",
      "enabled": true,
      "userRoles": [
        "string"
      ]
    }
  ]
}
```

## Get User

This request is used to get the details of the existing user.

**Method:** HTTP GET method

**Endpoint:** http://<host\_address>:<port\_address>/v1/user



*Get User Parameters*

Parameter	Cardinality	Description
X-API-AppId	Mandatory	The application ID is used for getting the user details. The default ID is auth.
X-API-Key	Mandatory	This key is used for getting the user details. The default ID is auth.
tenantId	Mandatory	This is the TENANT value as stored in the users table in the database.
userId	Mandatory	This is the username value as stored in the users table in the database.

## Delete User

This request is used to delete the existing user.

**Method:** HTTP DELETE method

**Endpoint:** `http://<host_address>:<port_address>/v1/user`

*Delete User Parameters*

Parameter	Cardinality	Description
tenantId	Mandatory	This is the tenant value as stored in the users table in the database.
X-API-AppId	Mandatory	The application ID is used for getting user details. The default ID is auth.
X-API-Key	Mandatory	This key is used for getting user details. The default ID is auth.
userInfo (Body)	userName Mandatory	It specifies the user name to be deleted.

Example for Delete User request:

```
[
  {
    "userName": "testuser",
  }
]
```

## Generating an authorization token

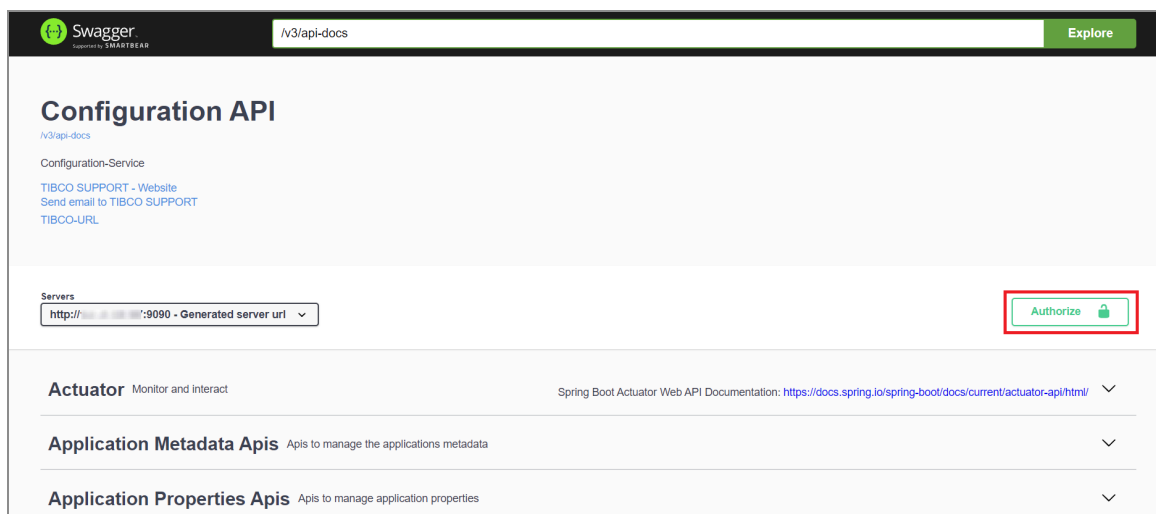
This token can be used to access the operations of all the services like data service, catalog service, orchestrator, and archival service.

### Procedure

1. To authorize a particular service, open the REST API home page of that service in a browser.

**Note:** If the `enableSecureAPI` value is set as `false`, the authentication is bypassed, and you do not have to authorize the service. For the REST services, the authorization token is not required. However, you must provide the `tenantID`.

2. Click the **Authorize** button.



The **Available authorizations** window opens.

## 3. Pass the following mandatory parameters:

*Authorization parameters and description*

Element Name	Element Type	Description
user name	String	username@tenantId
password	String	Existing password
Client credentials location		Select Authorization header or Request body from the dropdown options.
client_id	String	as provided in Tenant Registration
client_secret	String	as provided in Tenant Registration

**Available authorizations** X

Scopes are used to grant an application different levels of access to data on behalf of the end user.  
Each API may declare one or more scopes.  
API requires the following scopes. Select which ones you want to grant to Swagger UI.

**OAuth Password (OAuth2, password)**

Token URL: `http://.../9091/oauth/token`  
Flow: password

**username:**

**password:**

**Client credentials location:**

Authorization header ▾

**client\_id:**

**client\_secret:**

**Scopes:** [select all](#) [select none](#)

☐ read  
read scope

☐ write  
write scope

**Authorize** **Close**

4. Select the **read** and **write** checkboxes as per the requirements and then click the **Authorize** button.

## Result

An authorization token is generated for the particular service. This token is unique and valid only for the dedicated user with tenant ID. The access token comes with an expiry.

## Authorization Token APIs

- [Generate OAuth token](#)

**Note:**

- If you use an External Auth service, then User Management and Token Generation do not work. For this, use POSTMAN as Swagger authentication does not work.
- If you use Azure, the token is generated by Azure and not by Order Management Authorization service.
- OIDC works only with SSL.
- When you have chosen OIDC, Swagger cannot handle the OIDC flow.
- In the case of OIDC, if the token is expired, it generates an error.
- Client credentials cannot be handled via the Swagger.

## Generate OAuth Token

This request is used to generate authorization OAuth token.

**Method:** HTTP POST method

**Endpoint:** `http://<host_address>:<port_address>/oauth/token`

*Generate Authorization Header Parameters*

Parameter	Cardinality	Description
grant_type	Mandatory	You can select password or refresh token.
scope	Mandatory	You can select read, write, or 'read write'.
refresh_token		Refresh token from previously generated token. Required only when grant_type=refresh_token
user name		Required only when grant_type=password
password		Required only when grant_type=password

Parameter	Cardinality	Description
tenantId		Required only when grant_type=password
Authorization	Mandatory	
Content-Type	Mandatory	

## Configuring SSL for TIBCO Order Management

This section describes how to configure SSL for Order Management System; the web-based application components of TIBCO Order Management.

Configure SSL by using the following steps:

1. Edit the application.properties files in the following locations:

- <OM\_HOME>/roles/aopd/standalone/config/application.properties
- <OM\_HOME>/roles/archival-service/standalone/config/application.properties
- <OM\_HOME>/roles/authorization-service/standalone/config/application.properties
- <OM\_HOME>/roles/catalog-services/standalone/config/application.properties
- <OM\_HOME>/roles/configurator/standalone/config/application.properties
- <OM\_HOME>/roles/configurator-ui/standalone/config/application.properties
- <OM\_HOME>/roles/dataservice/standalone/config/application.properties
- <OM\_HOME>/roles/om-migration/standalone/config/application.properties
- <OM\_HOME>/roles/omsui/standalone/config/application.properties
- <OM\_HOME>/roles/orchestrator/standalone/config/application.properties
- <OM\_HOME>/roles/tmf-om-adapter/standalone/config/application.properties

- <OM\_HOME>/samples/processcomponent/standalone/config/application.properties

Add the following parameters to each application.properties file:

- server.ssl.key-alias=<key-alias>
- server.ssl.key-password=<key-password>
- server.ssl.key-store=classpath:<ssl-key-store-fileName>
- trust-store=classpath:<ssl-key-store-fileName>
- trust-store-password=<key-password>

Change the configuratorServiceUrl value to HTTPS url

2. Keep the keystore files in each directory or in as classpath resource.

- <OM\_HOME>/roles/aopd/standalone/config
- <OM\_HOME>/roles/archival-service/standalone/config
- <OM\_HOME>/roles/authorization-service/standalone/config
- <OM\_HOME>/roles/catalog-services/standalone/config
- <OM\_HOME>/roles/configurator/standalone/config
- <OM\_HOME>/roles/configurator-ui/standalone/config
- <OM\_HOME>/roles/dataservice/standalone/config
- <OM\_HOME>/roles/om-migration/standalone/config
- <OM\_HOME>/roles/omsui/standalone/config
- <OM\_HOME>/roles/orchestrator/standalone/config
- <OM\_HOME>/roles/tmf-om-adapter/standalone/config
- <OM\_HOME>/samples/processcomponent/standalone/config

3. Edit and save the files and then start the configurator UI.

- In a browser, open the following URL: [https://host:config-ui\\_port](https://host:config-ui_port)
- Log in to the configurator UI.
- On the **Order Management System UI** tab, under the **OMS UI Engine Configuration** app properties, update the

`com.tibco.af.omsui.httpChannelType` property value as 'HTTPS'.

4. Search for all services URLs and configure them for the HTTPS protocol.
5. Edit `<OM_HOME>/samples/processcomponent/standalone/config/application.properties`
  - Set `authorizationServiceTokenEndPoint` to HTTPS URL
  - Set `orchestratorBaseUrl` to HTTPS URL
6. Start (or restart) all the required services.

## Encrypt Password Utility

Encrypt Password Utility is an optional service. It is only used to encrypt a password.

1. In the `$OM_HOME/samples/EncryptPWDUtility/standalone/bin` directory, run `startEncryption.sh` (for Linux) or `startEncryption.ps1` (for Windows)

### Result

The password is encrypted or decrypted based on the API endpoint.



# TIBCO Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The documentation for this product is available on the [TIBCO® Order Management Product Documentation](#) page.

## How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

# Legal and Third-Party Notices

---

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, ActiveMatrix BusinessWorks, TIBCO Runtime Agent, TIBCO Administrator, and Enterprise Message Service are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2010-2024. Cloud Software Group, Inc. All Rights Reserved.