



Coupa Adapter for SAP Integration Suite

Version 1.0.3 – November 2025

Contents

1	Introduction	4
1.1	Objective	4
1.2	Coding Samples	4
1.3	Internet Hyperlinks.....	4
1.4	Overview.....	4
1.5	Features.....	5
2	Installation and Configuration.....	6
2.1	Package Details.....	6
2.2	Prerequisites	6
2.3	Adapter Installation on Cloud Foundry.....	6
2.3.1	Prerequisite.....	6
2.3.2	Procedure.....	7
2.3.2.1	Coupa Adapter Installation by Creating a New Integration Flow.....	7
2.3.2.2	Coupa Adapter Installation without Creating a New Integration Flow.....	7
2.3.3	Monitor the Deployment Status.....	8
2.4	Coupa Plug-in Installation.....	9
3	Getting Started: Coupa Adapter	11
3.1	Architecture Overview.....	11
3.2	Authentication	12
3.3	Supported Operations.....	13
4	Coupa Adapter Configuration.....	14
4.1	Receiver Adapter.....	14
4.1.1	General	14
4.1.2	Connection Tab	14
4.1.3	Processing Tab	15
5	Coupa Adapter Operations	19
5.1	Create Resource.....	19
5.1.1	Create Resource using Attachments	19
5.2	Get Resource By ID	21
5.3	Modify Resource.....	22
5.4	Query Resource.....	23
5.5	Update Resource	25

6	Payload and Dynamic Configuration.....	27
6.1	Message Payloads.....	27
6.2	Basic Dynamic Configuration Example.....	27
7	Eclipse Plug-in Configuration.....	29
7.1	XSD Generator.....	29
7.2	Attachment Response XSD's.....	30
8	Support.....	31
8.1	Traces.....	31
8.2	Troubleshooting.....	31
9	References.....	32
9.1	Initialize OAuth setup in Coupa and SAP CI.....	32
9.2	Initialize OAuth Setup for Coupa Adapter in SAP Integration Suite.....	34
9.3	Configure OAuth in Coupa Eclipse Plugin.....	34

1 Introduction

1.1 Objective

This is the official guide for Coupa Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Coupa adapter. Read this guide carefully before using the Adapter.

1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not guaranteed.

1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

1.4 Overview

Coupa is a unified cloud-based business spend management platform. Its module-based cloud platform makes it easy to cover, extend your financial management needs, and provides an overview of expenditures from everywhere. It efficiently captures, matches, and processes incoming expenses, aiming for a high accuracy rate (90%) in the initial matching step. This helps streamline expense management and reduces the need for manual adjustments.

The Coupa adapter helps you exchange data between Coupa and other systems. Some business scenarios that the adapter can be used for are as follows:

- Synchronizing master data.
- Synchronizing transactional data.
- Orchestrating the integration to realize a "Procure to Pay" process.


1.5 Features

The Coupa adapter has the following key features:

- **Secure authentication with OAuth 2.0:** Connecting Coupa with other applications is securely achieved by employing OAuth 2.0 authentication process. Every REST call between Coupa and SAP Integration Suite is secured by specific authorization flows.
- **Supports all Coupa operations:** Our Adapter has support for all Coupa operations including Create, Get, Modify, Query, and Update Resource operations.
- **Auto Pagination:** You can query efficiently with our integrated pagination support. Coupa adapter supports querying of all records of a resource using Auto Pagination.
- **Dynamic configuration with headers and properties:** Assigning dynamic values to different properties allows enhanced flexibility to your integration flows. You can also refer to dynamic parameters using SAP Integration Suite exchange headers and properties.
- **Support for XML & JSON:** Coupa Adapter extends support for multiple payload formats. XML and JSON are fully incorporated throughout the adapter.
- **Coupa Eclipse plug-in:** The Eclipse plug-in helps to generate all necessary up-to-date message XSDs for specific resources and the custom objects in the Coupa.
- **Full integration support for Custom Objects and Fields:** In addition to support for every standard Coupa resource, the adapter also integrates with all custom objects in Coupa.

2 Installation and Configuration

This section details the file(s) available as part of the installation package and lists the necessary prerequisites needed to install and configure the Coupa adapter.

 The Coupa adapter is available as part of your SAP Integration Suite license.

The Coupa adapter also comes with a plug-in to generate source and target XSDs. To use this plug-in, you need an Eclipse installation. For more information, see [Coupa Plug-in Installation](#).

Before you begin, you must download the Coupa plugin bundle from the SAP Marketplace that contains the file(s) listed in the following section.

2.1 Package Details

The Coupa plug-in installation package contains the following file:

File Type	File Name	Description
Plug-in File	<i>Coupa-plug-in_<version>.jar</i>	This file is used to install the plug-in.

2.2 Prerequisites

Before you begin, you must:

- Download and install the Coupa plugin bundle from the SAP Marketplace.
- Download and install the SAP-recommended version of Eclipse. Currently, Eclipse Oxygen is recommended.

For more information about the latest SAP recommendations see, [SAP Development Tools](#).

2.3 Adapter Installation on Cloud Foundry

Before the Coupa adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite.

2.3.1 Prerequisite

To deploy the Coupa adapter, you must have access to "Coupa Adapter for SAP Integration Suite" as part of your SAP Integration Suite license.

2.3.2 Procedure



The following installation procedures are compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

2.3.2.1 Coupa Adapter Installation by Creating a New Integration Flow

The Coupa adapter is available for selection in the receiver adapter list and can be deployed in the **Design** tab directly as you use it in an Integration flow.

Purpose

To install an adapter for use in your Integration flow.

Procedure

Go to **Design** workspace and select the integration package where you want to create a new Integration flow.

1. Click **Edit** to make the package editable.
2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.
3. Enter the **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
4. Go to the newly created integration flow and click **Edit** to make it editable.
5. In the integration flow, click **End** to add a **Connector** between the **End** and the **Receiver Box**. A drop-down with the available adapters appears. The **Coupa** adapter should show up in the list.
6. Select the **Coupa** adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once the Coupa Adapter is deployed, a success message is *displayed*.

After the above steps are done, the Coupa Adapter is successfully deployed in your Design workspace of the SAP Integration Suite tenant.

2.3.2.2 Coupa Adapter Installation without Creating a New Integration Flow



The following procedure describes how the Coupa adapter is migrated from the Discover workspace to the Design workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production.

The Coupa adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the Coupa adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported into the Design workspace by copying it from the Discover workspace.

Purpose

To import the Coupa adapter to **Design** workspace by copying the integration package from **Discover** workspace.

Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **Coupa adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**.
This copies the package from the Discover workspace to Design workspace.
4. Go to Design workspace and select the copied **Coupa adapter for SAP Integration Suite** package.
5. In the **Actions** tab of the selected package, click **Deploy**.
This completes the adapter deployment to Design workspace.

2.3.3 Monitor the Deployment Status

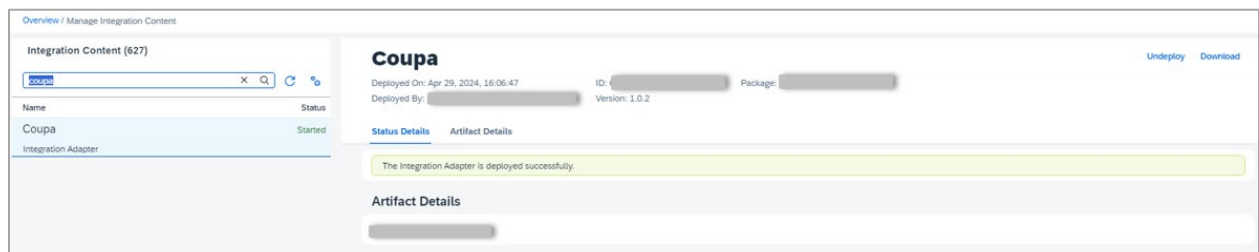
After the adapter deployment is complete, you can check the status in the **Monitor** section.

Purpose

To check the status of the deployed adapter.

Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.



2.4 Coupa Plug-in Installation

Prerequisites

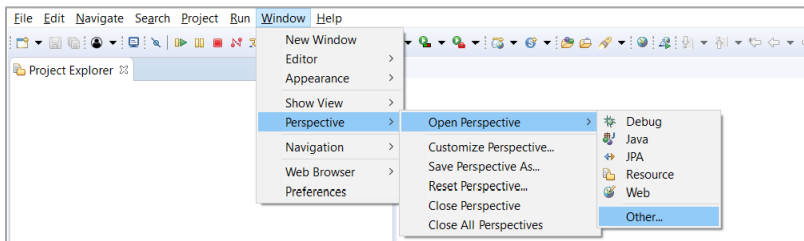
Before you begin, you must download the Coupa plugin bundle from the SAP Marketplace. For more information, see [Coupa Installation and Configuration](#).

Purpose

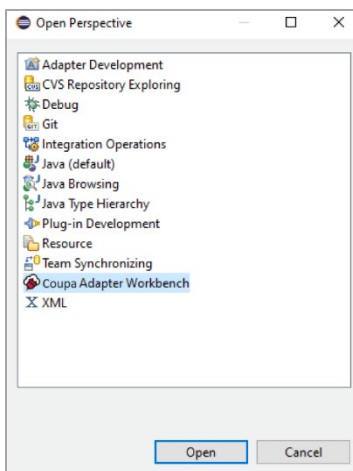
To install the Coupa plug-in.

Procedure

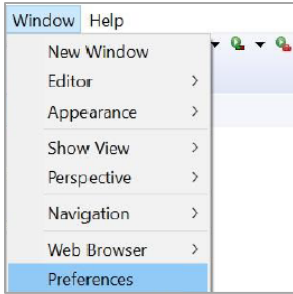
1. Download and install Eclipse Oxygen 3a from [Eclipse Oxygen](#).
2. After Eclipse installation, go to your Eclipse installation folder (Example: C:\Program Files (x86)\oxygen\eclipse) and open the Plug-in folder.
3. Copy the Coupa Plug-in JAR file (that you downloaded from the SAP Marketplace) into the Plug-in folder.
4. Open Eclipse Oxygen and go to **Windows > Perspective > Open Perspective > Other**.



5. Select **Coupa Adapter Workbench** and click **Open**.



6. Go to **Window > Preferences > Coupa Preferences**.

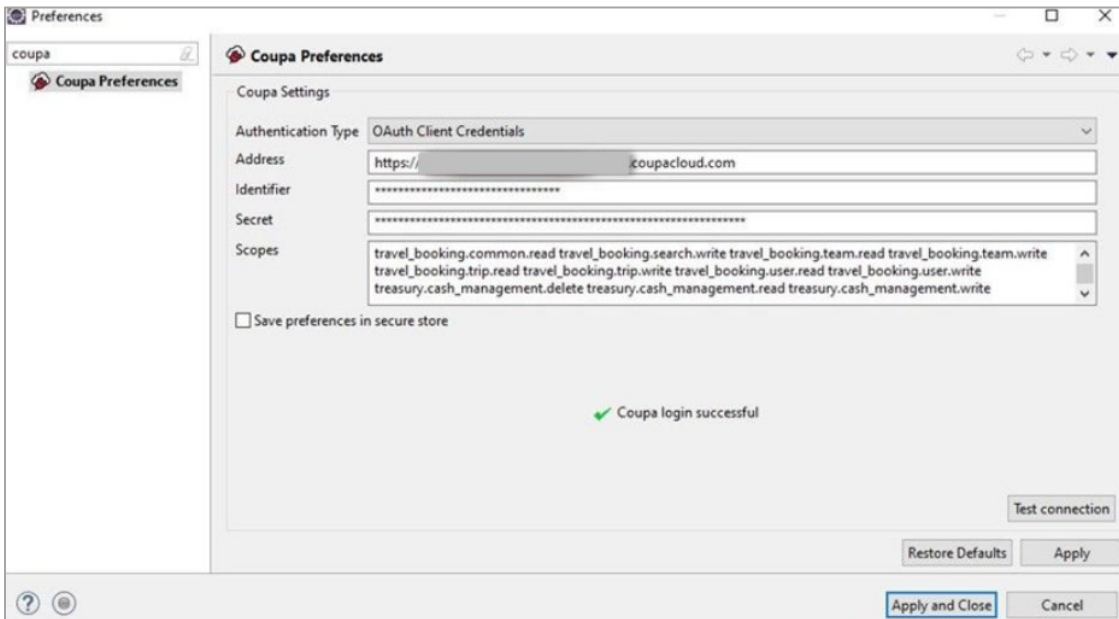


7. In the **Coupa Preferences** widget, fill in the **Coupa Settings** details. You can retrieve these details from your Coupa tenant.

i To retrieve Settings details and respective login credentials to set up a Coupa connection to the specific tenant via an OAuth 2.0 app, see [Coupa Adapter Guide](#).

8. Verify the Settings details and the credentials by clicking the **Test Connection** tab.
9. Upon successful test, click **Apply and Close** to complete the installation.

i After you hit **Apply and Close**, the plugin takes a while to set up based on your saved preferences. The required swagger files are being retrieved from the API documentation in the background. Consequently, the loading time varies depending on your machine.



Now you are set up to use your Coupa adapter and Coupa plug-in.

3 Getting Started: Coupa Adapter

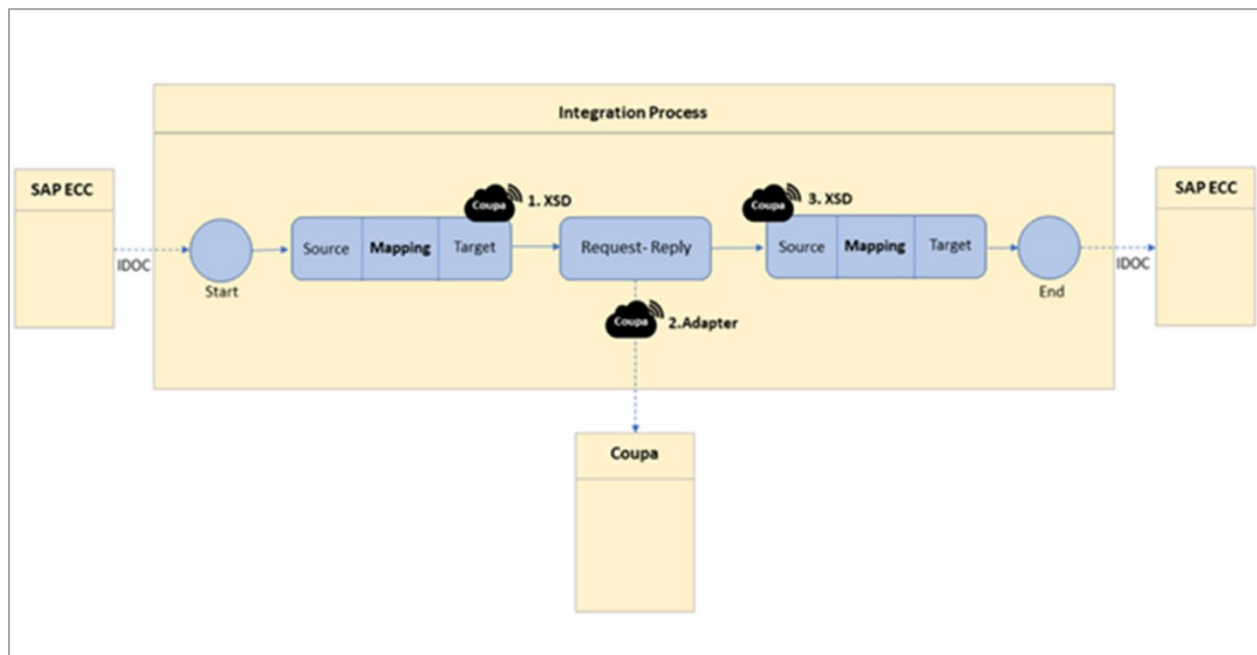
Before you start using the adapter, you can get to know the underlying [Architecture](#) for the adapter and [Authentication](#) mechanism for Coupa. Finally, a list of all the [Supported Operations](#) have been provided. Comprehensive details of the operations can be found in the [Coupa Adapter Operations](#).

3.1 Architecture Overview

The Coupa adapter can be employed as a receiver adapter. In such a scenario where the Coupa Adapter is used as a receiver adapter, SAP Integration Suite acts as the initiator of the calls.

The Coupa adapter supports various operations such as Create Resource, Get Resource by ID, Modify Resource, Query Resource, and Update Resource. To know more about the different operations supported, see [Supported Operations](#).

The figure below demonstrates the request-reply step using a Coupa adapter and gives a high-level representation of how the adapter works.



The Coupa adapter works as a *Request-Reply* step in the integration process, responsible for connecting and interacting with Coupa to invoke its operations.

The Eclipse Plug-in, which is included in the adapter package, is used to help the user to generate XSDs of the different entities or resources that are available in your Coupa tenant.

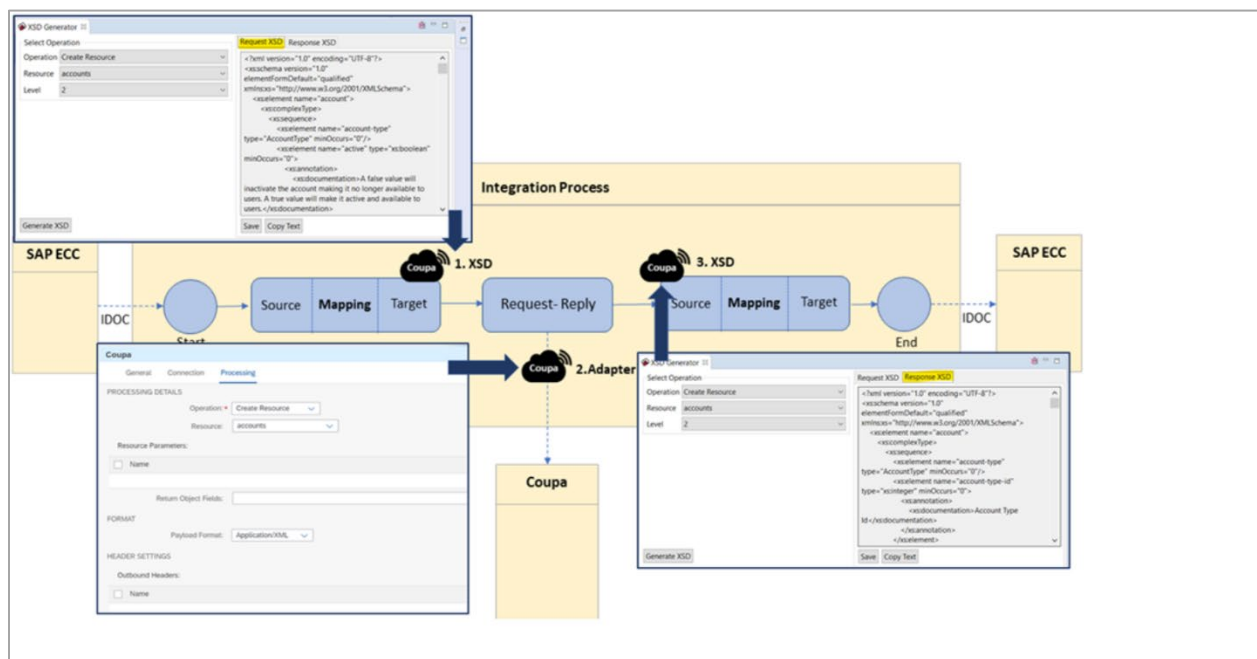
The XSDs generated by the Eclipse Plug-in can be imported in an Integration Flow and used in mappings.

- For instance, it can be added before the *Request-Reply* Step in the above mapping. This way, you can create the correct XML request message to create or modify a resource in Coupa (see the Mapping Step on the left side of the Request-Reply Step).
- Similarly, you can use the XSDs to create a mapping to handle the response message returned by Coupa (see the Mapping Step on the right side of the Request-Reply Step).

Furthermore, the Eclipse Plug-in XSD Generator helps you generate up to date XSDs for your Coupa resources for all the different operations like create, get, and modify.

For more details about the Coupa plug-in and XSD Generator configuration, see [Coupa Plug-in Installation](#).

The figure below portrays the stages of Coupa adapter usage in the SAP Integration Suite flow, including Coupa Plug-in generated request XSD and response XSD.



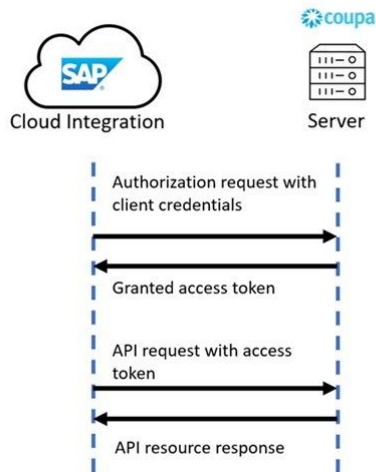
3.2 Authentication

The Coupa adapter supports only OAuth 2.0 authentication. The adapter uses common security artifacts stored in SAP Integration Suite. You must use Secure Parameter and User Credentials to safely store OAuth tokens, user-password combinations, and secrets. These security artifacts can then be accessed in the adapter using aliases.

The Coupa adapter uses the OAuth 2.0 Autonomous Client type. The adapter requires OAuth Client Credentials with Client Identifier, Client Secret, and Scope values.

These app credentials consist of a client id and client secret. The Coupa adapter can make different requests to Coupa resources using the access token that is returned from the authorization server.

The diagram below explains the process used in the background to connect to Coupa.



3.3 Supported Operations

The operations supported by the Coupa adapter are listed below:

- [Create Resource](#)
- [Get Resource by ID](#)
- [Modify Resource](#)
- [Query Resource](#)
- [Update Resource](#)

Create Resource is selected by default and allows you to integrate with Coupa using XML or JSON formats. The integration is carried out securely by utilizing the OAuth protocol for authenticating your REST calls. All the operations enable the user to access the full data resources of Coupa and execute CRUD operations.

4 Coupa Adapter Configuration

This section describes the parameters to be configured for your Coupa adapter. You need to configure the **Connection** and **Processing** tabs. A description and example usage for every field has been added.

4.1 Receiver Adapter

In this section, you will learn how to configure the Coupa receiver adapter. On selecting the Coupa adapter from the list of adapters, you must configure the **General**, **Connection**, and **Processing** tabs.

4.1.1 General

The General tab provides an overview of basic adapter information including **Channel** and **Adapter** details.

The screenshot shows the configuration interface for a Coupa adapter. At the top, there are tabs for 'General', 'Connection', and 'Processing', with 'General' selected. Below the tabs, there is a 'Name' field containing 'Coupa'. The interface is divided into two main sections: 'CHANNEL DETAILS' and 'ADAPTER DETAILS'. Under 'CHANNEL DETAILS', there are three fields: 'Direction' set to 'Receiver', 'System' set to 'Receiver', and 'Description' set to 'Coupa integration flow1'. Under 'ADAPTER DETAILS', there are three fields: 'Adapter Type' set to 'Coupa', 'Transport Protocol' set to 'HTTPS', and 'Message Protocol' set to 'REST'. The interface also includes an 'Externalize' button and some utility icons in the top right corner.

Field	Description
Name	Specify Name for your Coupa Adapter.
Description	Specify the required description.


4.1.2 Connection Tab

The Connection tab contains connection and authentication parameters for Coupa. The adapter only supports OAuth 2.0 authentication.

A sample configuration for Connection Tab is shown in the image below.

The screenshot shows the 'Coupa' configuration page with the 'Connection' tab selected. Under 'CONNECTION DETAILS', there is a dropdown for 'Authentication' set to 'OAuth Client Credentials'. Below it are input fields for 'Address', 'OAuth Credential Name', and 'Scopes'. The 'Scopes' field contains a list of permissions: 'core.accounting.read core.accounting.write core.approval.read core.approval.write core.budget.read core.budget.write core.business_entity.read core.business_entity.write'. At the bottom, there are two fields for timeouts: 'Connection Timeout (in ms): 60000' and 'Response Timeout (in ms): 60000'.

The connection tab contains the following fields:

Field	Description
Address	Specify the recipient's endpoint URL, this is Coupa tenant's URL of the organization. Example: <code>https://<your_coupa_instance>.coupacloud.com</code>
OAuth Credential Name	Specify the name of the User Credentials artifact configured or saved in the SAP Security Material. It contains the Coupa OAuth key and client secret pair.  This property enables the system to fetch the security token from the Keystore for authentication.
Scopes	Specify the permissions configured in Coupa. You can enter multiple scope values separated by a space. Example: <code>core.accounting.read core.account.write core.budgeting.write</code>
Connection Time Out (in ms)	Specify the connection timeout in milliseconds. You can configure the maximum waiting time for the connection to be established between SAP and Coupa. Example: 3000
Response Timeout (in ms)	Specify the response timeout in milliseconds. You can configure the maximum waiting time for SAP until a response is received from Coupa. Example: 3000


For more information on how to retrieve details of the Coupa OAuth key-client secret pair, see [References](#).




4.1.3 Processing Tab

The Processing tab contains all the operational configurations for the Coupa adapter.

A sample configuration for the operation Create Resource (Accounts) is shown in the image below.

The processing tab contains the following fields:

Field	Description
Operation	<p>Specify the type of action you want to run in Coupa. To access and exchange data with Coupa, you can choose one of the available operations.</p> <p>For more information, see Supported Operations.</p>
Resource	<p>Select the name of the Coupa resource to perform the API operation. It can be any of the Coupa defined resources like <i>Reference Data</i>, <i>Transactional</i>, <i>Shared</i> resources, or user defined custom object resources. This is an editable dropdown field; you can manually add any other resource that are not listed in the dropdown.</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Since the dropdown is editable, ensure you do not leave the operation field blank.</p> </div>
Resource Parameters	<p>Specify the resource key and value in case resource path includes parameters.</p> <p>Syntax: If <code>Account/:id</code> is the resource, Name should be specified with <code>id</code>, and Value should be specified with <code><account_id> value</code> the corresponding value pair.</p> <p>Example: Name: <code>id</code>, Value: <code>6351</code></p>
Return Object Fields	<p>Specify the set of fields to be passed in the response payload. The format of the "return objects" must be JSON.</p> <p>Example: <code>["firstname", "lastname", {"roles": ["id", "name"]}]</code></p>
Payload Format	<p>Specify the format of the request message to be sent to and the response to be returned from Coupa.</p> <p>Acceptable Values: <code>Application/JSON</code>, <code>Application/XML</code></p>

Field	Description
<p> Create Resource operation allows working with attachments. After selecting a resource including attachments such as <code>invoices/:invoice_id/attachment</code>, additional fields appear as described below. For more information, see Create Resource using Attachments.</p>	
Data from Attachment	<p>Select this checkbox to specify whether the attachment object should be obtained from an attachment of the Message Exchange in SAP CI or the from the Body in SAP CI.</p>
Attachment name	<p>When selecting data from attachment, specify the name of the attachment on the exchange in SAP CI to retrieve the file.</p> <p>Example: <code>Filename.zip</code>, <code>\${property.filename}.zip</code></p> <p>This field appears when you check Data from Attachment checkbox.</p>
File Name	<p>Specify the file name to be uploaded to Coupa.</p>
Type	<p>The type of attachment for Coupa.</p> <p>Default: <code>file</code>.</p>
Attachment Properties	<p>Specify the key and value in case attachment includes parameters. You can add Property Name and Property Value pairs for various fields like <code>file-url</code>, <code>linked-to</code>, <code>intent</code>, <code>text</code>, <code>url</code>.</p>
<p> Query Resource operation allows querying of any Coupa Resource. After selecting this operation additional fields appear that help you to customize your query requests. For more information, see Query Resource.</p>	
Limit	<p>Specify the number of records to be fetched with a maximum of 50.</p> <p>Example: <code>20</code></p> <p> When auto pagination is enabled, the Limit value does not work as all the records will be queried or up to the pre-defined limit value of 100.</p>
Offset	<p>Specify the offset value to be used.</p> <p>Example: <code>20</code></p>
Auto Pagination	<p>Select the checkbox to enable retrieval of all records in case query result has more than one page of records.</p>

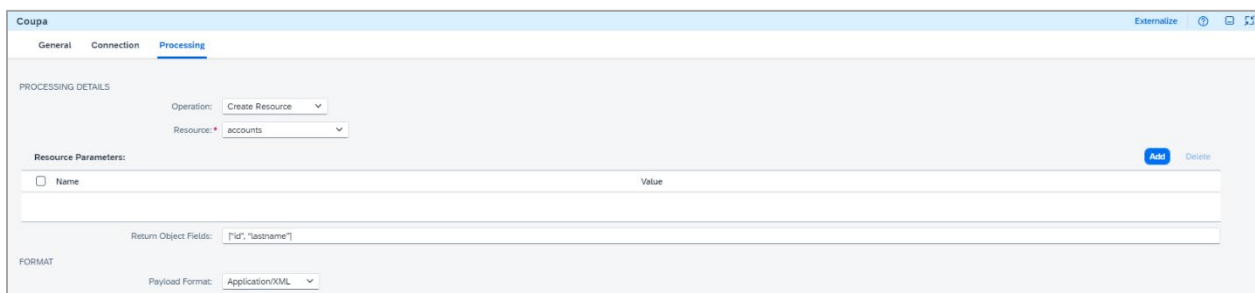
Field	Description
Query Filter Parameter	<p>Specify the querying parameters to retrieve the records. Name, operator, and value can be added for each filtering parameter.</p> <p>Example:</p> <p>Name Contains Created date</p> <p>Operator Contains Greater Than</p> <p>Value Contains 2202-01-01</p>
Order by Field	<p>Specify the field or resource property to perform the sorting of the queried records.</p> <p>Example: Name</p>
Sorting Direction	<p>Specify the sorting order of the records. It can be either Ascending or Descending.</p> <p>Acceptable Values: Ascending, Descending</p>

5 Coupa Adapter Operations

This section contains a comprehensive description of all the operations supported by the Coupa adapter.

5.1 Create Resource

This operation allows creation of any new record of a specified Coupa resource. The request payload should be provided with all the required property and field values of the resource to be created in the request. The response body will contain the unique identifier (ID) of the created record if the call is successful and returns all the Coupa elements of the resource.



The screenshot shows the 'Coupa' configuration window with the 'Processing' tab selected. Under 'PROCESSING DETAILS', the 'Operation' dropdown is set to 'Create Resource' and the 'Resource' dropdown is set to 'accounts'. Below this, there is a 'Resource Parameters' section with a table for adding parameters. The 'Return Object Fields' field contains the value ['id', 'lastname']. At the bottom, the 'Payload Format' dropdown is set to 'Application/XML'.

The configuration details are as follows:


Field	Description
Operation	Specify the operation to be performed: <code>Create Resource</code> .
Resource	Specify any of the available resource types in Coupa - Reference Data resource, Transactional resource, Shared Resource or Custom object resource. Example: <code>accounts</code>
Return Object Fields	Specify only limited elements to be returned in the response payload. Example: <code>["id", "lastname"]</code>
Payload Format	Specify the request and response payload format. Acceptable Values: <code>Application/JSON</code> , <code>Application/XML</code>

5.1.1 Create Resource using Attachments

The Create Resource operation also allows you to work with attachments. After selecting a resource that includes attachments such as `invoices/:invoice_id/attachment`, additional options appear in the **Processing** tab.

The screenshot shows the 'Coupa' interface with the 'Processing' tab selected. Under 'PROCESSING DETAILS', the 'Operation' is set to 'Create Resource' and the 'Resource' is 'invoices/invoice_id/att...'. The 'Attachment Name' is 'File123.zip' and the 'File Name' is '\${property.filename}.zip'. The 'Type' is set to 'file'. There are also sections for 'Resource Parameters', 'Attachment Properties', and 'FORMAT'.

The configuration details are as follows:

Field	Description
Operation	Specify the operation to be performed: <code>Create Resource</code> .
Resource	Specify any of the available resources in Coupa which includes an attachment. Example: <code>invoices/:invoice_id/attachments</code>
Data from Attachment	Select this checkbox if the attachment object must be obtained from an attachment of the Message Exchange in SAP CI. Leave the checkbox unchecked if the attachment object is being retrieved from the Body in SAP CI.
Attachment Name	When selecting data from attachment, specify the name of the attachment on the exchange in SAP CI to retrieve the file.  This value can be read dynamically using an exchange header or property. <i><code>\${property.filename}.zip</code></i> Example: <code>Filename.zip, \${property.filename}.zip</code>
File Name	Specify the file name to be uploaded to Coupa.
Type	The type of attachment for Coupa. The dropdown allows selection from multiple options including <code>file</code> , <code>text</code> , and <code>url</code> . Default value: <code>file</code>
Attachment Properties	You can add Property Name and Property Value pairs for various fields like <code>file-url</code> , <code>linked-to</code> , <code>intent</code> , <code>text</code> , and <code>url</code> .

Field	Description
Payload Format	Specify the request and response payload format. Acceptable Values: Application/JSON, Application/XML

The below table elaborates the available option for **Attachment Properties**.

Property Name	Property Value
Intent	A label that can be added to the attachment, that is visible in Coupa. Example: Supplier
file-url	URL to the attached file to be sent to Coupa with the attachment.
linked-to	Link to a specific feature to be sent to Coupa with the attachment.
Text	Text element to be sent to Coupa with the attachment in case Type was selected as text.
url	URL element to be sent to Coupa with the attachment, in this case Type was selected as url.

5.2 Get Resource By ID

The operation allows retrieval of a specified record of a Coupa Resource by ID. The ID of the resource can be provided in the resource parameter or can be passed as a dynamic header value. The response body returns all the Coupa elements of the specified target resource.

The screenshot shows the 'Coupa' interface with the 'Processing' tab selected. The 'Operation' is 'Get Resource By ID' and the 'Resource' is 'accounts/id'. The 'Resource Parameters' table has one row with 'id' and value '1653'. The 'Return Object Fields' are '[email], [lastname]'. The 'Payload Format' is 'Application/XML'.

The configuration details are as follows:

Field	Description
Operation	Specify the operation to be performed: <code>Get Resource by ID</code> .
Resource	Specify the target resource to be fetched. Example: <code>accounts/:id</code>
Resource Parameters	Specify the resource id or any other element as key and value pair. Example: Name: <code>id</code> , Value: <code>1653</code>
Return Object Fields	Specify only limited elements to be returned in the response payload. Example: <code>["id", "lastname", "email"]</code>
Payload Format	Specify request and response payload format. Acceptable Values: <code>Application/JSON</code> , <code>Application/XML</code>

5.3 Modify Resource

The operation allows modification of any specific record of a Coupa Resource by passing the ID. The ID of the resource can be provided in the resource parameter or can be passed as dynamic header value.

This operation uses *Patch API* of Coupa. After updating the record in Coupa, the response body returns all the Coupa elements of the specified resources that are being updated.

The screenshot shows the 'Coupa' configuration tool interface. The 'Processing' tab is active, displaying the following configuration details:

- OPERATION:** Modify Resource (selected from a dropdown)
- RESOURCE:** accounts/id (selected from a dropdown)
- RESOURCE PARAMETERS:** A table with columns 'Name' and 'Value'. One parameter is defined: 'id' with value '6351'. There are 'Add' and 'Delete' buttons.
- RETURN OBJECT FIELDS:** ["id", "status", "website"]
- FORMAT:** Payload Format: Application/XML (selected from a dropdown)
- HEADER SETTINGS:** A table with columns 'Name' and 'Value'. It is currently empty, with 'Add' and 'Delete' buttons.

The configuration details are as follows:

Field	Description
Operation	Specify the operation to be performed: <code>Modify Resource</code> .
Resource	Specify the target resource to be modified. Example: <code>accounts/:id</code>
Resource Parameter	Specify resource id or any other element as key and value pair. Example: Name: id, Value: 1653
Return Object Fields	Specify required elements to be returned in the response payload. Example: <code>["id", "status", "website"]</code>
Payload Format	Specify request and response payload format. It can be either XML or JSON. Acceptable Values: <code>Application/JSON</code> , <code>Application/XML</code>

5.4 Query Resource

The operation allows querying of any Coupa Resource. You can customize your query requests by configuring **Query Filter Parameters**, **Order by Field** elements, and **Sorting Direction** (Ascending or Descending).

The response body will return all the Coupa elements of the specified resource that are being queried, you can also **Limit** the maximum number of records to be fetched, set the **Offset** for the fetching records, and enable **Auto Pagination** feature for bulk records retrieval.

The configuration details are as follows:

PROCESSING DETAILS

Operation: Query Resource
Resource: accounts

Resource Parameters:

Name	Value
id	6351

Return Object Fields: ["id", "name", "email"]
Limit: 20
Offset: 20
Auto Pagination:

Query Filter Parameters:


Name	Operator	Value
Name	Contains	Created date
Operator	Contains	Greater Than
Value	Contains	2022-01-01

Order By Field:

Field
Name

Sorting Direction: Ascending

A description of the fields is given in the table below.

Field	Description
Operation	Specify the operation to be performed Query Resource.
Resource	Specify the target resource to be queried. Example: <code>accounts</code>
Resource Parameters	Specify resource id or any other element as key and value pair. Example: Name: id, Value: 6351
Return Object Fields	Specify required elements to be returned in the response payload. Example: <code>["id", "name", "email"]</code>
Payload format	Specify request and response payload format. It can be either XML or JSON. Example: <code>application/XML</code>
Limit	Specify the limit to query records with maximum of 50 records. Example: 20 <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;">  When auto pagination is enabled, the Limit value does not work as all the records will be queried or up to the pre-defined limit value of 100. </div>
Offset	Specify the offset value for the number of records to be skipped while querying. Example: 20
Auto Pagination	Select the checkbox to enable retrieval of all records in case query result has more than one page of records.

Field	Description
Query Filter Parameter	Specify the querying parameters to retrieve the records. Name , Operator , and Value can be added for each filtering parameter. Example: Name Contains Created date Operator Contains Greater Than Value Contains 2202-01-01
Order by Field	Specify the field or resource property to perform the sorting of the queried records. Example: Name
Sorting Direction	Specifies the sorting order of the records. It can be either Ascending or Descending. Acceptable Values: Ascending, Descending Default: Descending

5.5 Update Resource

This operation allows update of any specific record of a Coupa Resource by passing the ID of the resource. You can provide the ID in the **Resource** parameter or pass it as a dynamic header value. This operation uses PUT API of Coupa. After updating the record in Coupa, the response body returns all the Coupa elements of the specified resource.

The screenshot displays the 'Coupa' API interface, specifically the 'Processing' tab. The 'Operation' is set to 'Update Resource' and the 'Resource' is 'suppliers/id'. Under 'Resource Parameters', the 'id' field is populated with '1536'. The 'Return Object Fields' are set to '["id", "status", "website"]'. The 'Payload Format' is 'Application/XML'. The 'Outbound Headers' section is currently empty.

The configuration details are as follows:

Field	Description
Operation	Specify the operation to be performed: Update Resource.
Resource	Specify the target resource to be updated. Example: <code>supplier/:id</code>
Resource Parameters	Specify resource id or any other element as key and value pair. Example: Name: id, Value: 1536
Return Object Fields	Specify required elements to be returned in the response payload. Example: <code>["id", "status", "website"]</code>
Payload Format	Specify request and response payload format. It can be either XML or JSON. Example: <code>application/XML</code>

6 Payload and Dynamic Configuration

The Coupa adapter supports different message formats. It also supports dynamic configuration of its properties using both the header and property features of the SAP Integration Suite exchange. Both header and property features of the exchange can be used in Coupa adapter properties.

6.1 Message Payloads

Many operations require a Message as input to be sent to Coupa. For example, to create an object in Coupa using the Create Resource operation, you need to specify the fields and values in the message body to be inserted in Coupa. To achieve this, you first need to specify the Coupa Resource type in the Coupa adapter.

An example payload to create an Address is shown in the image below:

```
<?xml version="1.0" encoding="UTF-8"?>
<address>
  <name>Address Name</name>
  <city>San Mateo</city>
  <postal-code>989048</postal-code>
  <state>CA</state>
  <street1>2 W 5th Ave</street1>
  <street2>Suite 300</street2>
  <country>
    <code>US</code>
    <name>United States</name>
  </country>
</address>
```

The XSD Generator assists in determining what structure each Coupa adapter operation expects. It generates the XSDs that you can use in mappings in your integration scenario. Following our example, the XSD will include elements/properties of the Objects.

Assuming that you are performing a Create Resource operation in Coupa, using the correct request XSD as the target of the mapping before the request-reply will help you create a valid input message (payload) towards Coupa.

The XSDs obtained from the Eclipse Plug-in can help to provide a good understanding of the request message that the various Coupa operations require as well as the response messages that you can receive from Coupa.

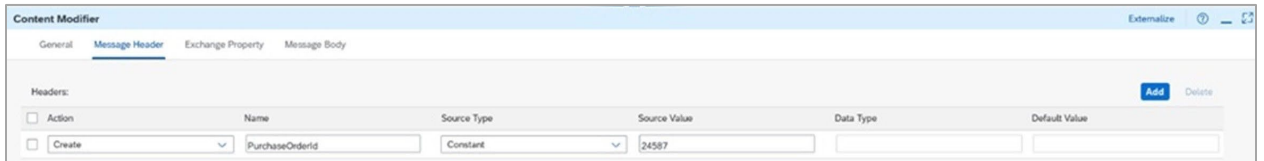
6.2 Basic Dynamic Configuration Example

Purpose

This section illustrates the use of dynamic values in the adapter property. It depicts a simple Get Resource by ID operation configuration using a dynamic header reference.

Procedure

1. You need to use a content modifier before the Request-Reply step in your integration flow, and this content modifier should create a **Message Header**. The name of the message header is the name to be referred to in the Coupa adapter. Any name is allowed here. You can achieve the same by using the properties instead of a header. In this example, we use the Constant type and fill in the Purchase Order Id as a value.

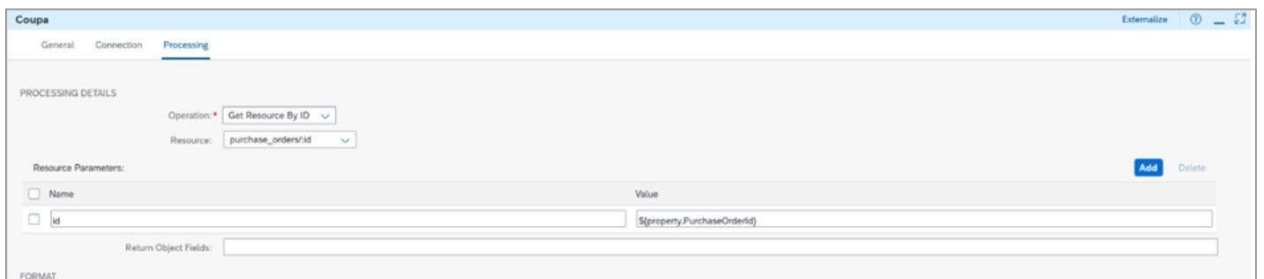


The screenshot shows the 'Content Modifier' interface with the 'Message Header' tab selected. A table lists headers with columns for Action, Name, Source Type, Source Value, Data Type, and Default Value. One header is configured with Action 'Create', Name 'PurchaseOrderId', Source Type 'Constant', and Source Value '24587'.

Action	Name	Source Type	Source Value	Data Type	Default Value
Create	PurchaseOrderId	Constant	24587		

2. The Coupa Receiver adapter (which is linked to the Request-Reply step) needs to refer to this header by using **\${header.name}**. To refer to a property, a syntax similar to **\${property.name}** should be used.

In our example below, we can refer to the `PurchaseOrderId` header using the syntax **\${header.AccountId}**. The Coupa Adapter will automatically replace this reference with the ID 24587.



The screenshot shows the 'Coupa' interface with the 'Processing' tab selected. Under 'PROCESSING DETAILS', the 'Operation' is 'Get Resource By ID' and the 'Resource' is 'purchase_orders/id'. Under 'Resource Parameters', a parameter named 'id' is configured with the value `$(property.PurchaseOrderId)`.

Name	Value
id	\$(property.PurchaseOrderId)

3. Select the level in the **Level** dropdown.
4. Click **Save** to save the displayed XSD, switch tabs to save the corresponding response/request and click **Copy Text** to copy the XSD text to your clipboard.

The below table outlines the Request/Response XSDs generated for Coupa operations specific to a Coupa object.

Operation	Object Specific	Request	Response
Create Resource	X	X	X
Get Resource by ID	X		X
Modify Resource	X	X	X
Query Resource	X		X
Update Resource	X	X	X

7.2 Attachment Response XSD's

Response XSD's for attachment related resources contain different operation elements like attachment-file, attachment-URL and attachment-text. When using these XSDs in the Message Mapping UI, it requires a selection of one of these elements to support the attachment type such as file, URL or text based on what has been configured in the adapter.

Resources for which this applies include the resources with attachment in the name.

Example: `contracts/:contract_id/attachments.`

Select An Element

attachment-file

attachment-url

attachment-text

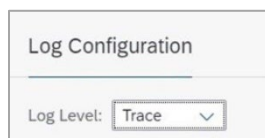
Cancel

8 Support

8.1 Traces

You can monitor, debug, and analyse errors or issues by changing the **Log level** of your integration flow to **Traces**.

You can enable this under **Log Configuration** under **Monitor > Manage Integration Content** page in SAP Integration Suite.



The Trace Log level enables the collection of more traces to effectively understand the issues. These traces can also be used in a ticket.

8.2 Troubleshooting

A few issues that you might encounter and possible solutions for them have been documented below.

Error	Error Message	Possible Solution
No Artifact Descriptor Found	[CAMEL][IFLOW][CAUSE]: Cause: com.sap.it.nm.types.NodeManagerException: [CONTENT][CONTENT_DEPLOY][NoArtifactDescriptorFoundForArtifactName]:No artifact descriptor found for artifactName <CoupaOauth>	This error generally indicates that one of the security artifacts used in the Connection tab in the adapter does not exist. Ensure that the artifact used in the connection tab exists as a Security Material.

Error	Error Message	Possible Solution
XSD Generation not supported for this resource	<code><!-->XSD generation not support for this resource<--></code>	When in the Eclipse Plug-In, this error is visible after trying to generate a request or response XSD, this means that a part of the structure for the resource that you have selected on the left side is not currently supported.

9 References

This section provides a step-by-step guide for setting up OAuth authorization for different platforms like Coupa and SAP CI, Cloud Platform Coupa Adapter, and Eclipse Coupa Plugin.

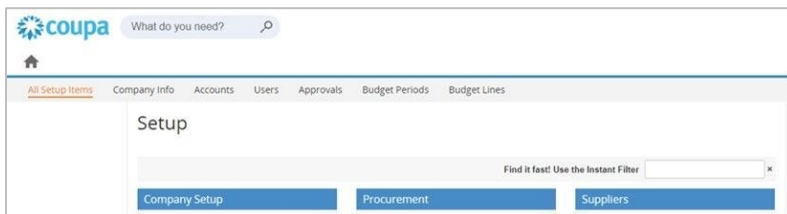
9.1 Initializing OAuth setup in Coupa and SAP Integration Suite

Purpose

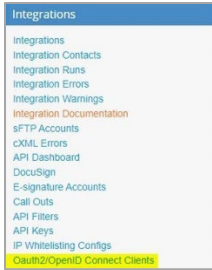
To initialize OAuth setup in Coupa and SAP integration Suite.

Procedure

1. Login to the Coupa Tenant using the Coupa URL for the organization.
2. Click **All Setup Items**.



3. Go to **Integrations > Oauth2/OpenID Connect Clients**.



4. Click **Create**, enter the details as required, select appropriate permissions, and click **Save**.

5. On saving, you can view the below details. Save this information to connect to Coupa externally.

The **Oidc Scopes** are the permissions configured in Coupa. You can use these values (separated by a space) in the **Scopes** field of the Coupa adapter while configuring the **Connection** tab.

Furthermore, **Identifier** and **Secret** are required in an upcoming section. For more information, see [Configure OAuth in Eclipse Coupa Plugin](#).

9.2 Initializing OAuth Setup for Coupa Adapter in SAP Integration Suite

Purpose

To initialize OAuth Setup for Coupa Adapter in SAP Integration Suite.

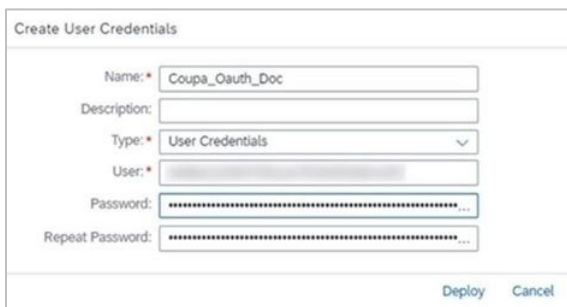
Procedure

1. Open **Monitor** tab in SAP Integration Suite.
2. Select **Security Material** from the **Manage Security Material** section.



Name	Type	Status	Deployed By	Deployed On
------	------	--------	-------------	-------------

3. Select **Add, User Credentials**, and enter a **Name** which is same as the **Basic Credential Name** that you have configured in the Coupa adapter **Connection** tab. In User and Password, insert **User** with the **Coupa Identifier** and **Password** with the **Coupa Security Key**. Finally, Click **Deploy**.



Create User Credentials

Name: * Coupa_Oauth_Doc

Description:

Type: * User Credentials

User: *

Password: *

Repeat Password: *

Deploy Cancel

The Coupa OAuth setup is successfully deployed for a secure OAuth connection to Coupa.



It is important that the name configured for the OAuth Credential Name in the adapter matches the names of the deployed *artifacts* in the SAP secure stores as described in the steps above.

Now that the authentication part is completed, the Coupa adapter can be configured for every specific Coupa integration scenario.

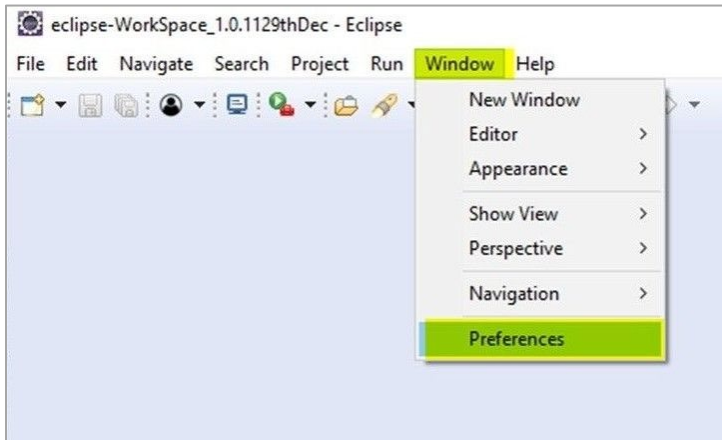
9.3 Configure OAuth in Coupa Eclipse Plugin

Purpose

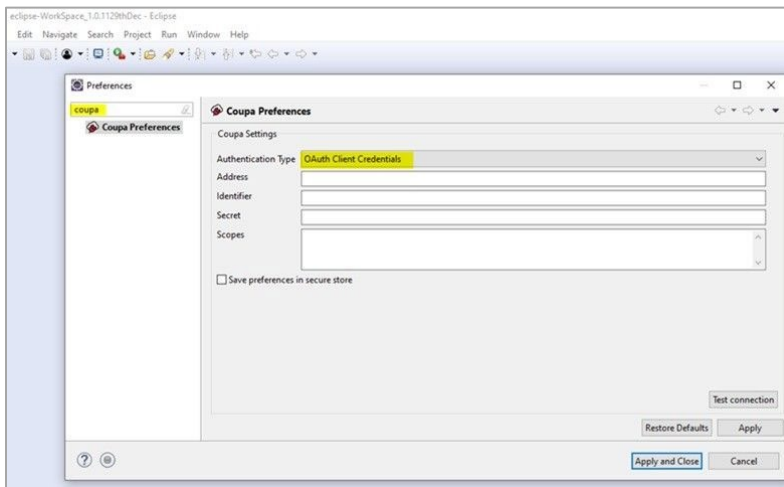
To Configure OAuth in Coupa Eclipse Plugin.

Procedure

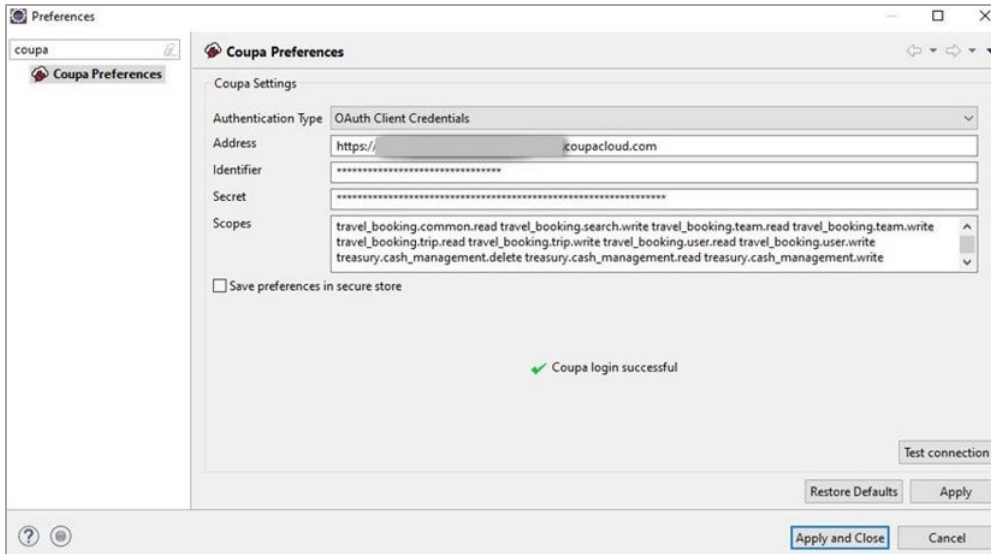
1. To install Coupa Plugin in Eclipse, see [Coupa Plug-in Installation](#).
2. Once the Plugin is Installed, Open **Eclipse** > **Windows** > **Preferences**. This will open the Preferences Window.



3. From the List of **Preferences** on the left, filter for Coupa.



4. **Authentication type** dropdown defaults to OAuth Client Credentials.
5. Enter the **Address** of the Coupa Tenant.
6. For **Identifier**, **Secret** and **Scope** values, enter the values generated from the Coupa Tenant.
7. Click **Test Connection**.



8. Once the account validation is **successful**, check the box **Save Preferences** in the secure store and click **Apply and Close**.



After you hit **Apply and Close**, the plugin takes a while to set up based on your saved preferences. The required swagger files are being retrieved from the API documentation in the background. Consequently, the loading time varies depending on your machine.