# Oracle® Cloud

# Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3

F45605-05
December 2025

Oracle Cloud Using the SAP Commerce Cloud (Hybris) Adapter with Oracle Integration 3,

F45605-05



Primary Author: Oracle Corporation

# Contents

# About This Content

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

### Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

### Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

### Related Resources

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

### Conventions

The following text conventions are used in this document.

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

1

# Understand the SAP Commerce Cloud (Hybris) Adapter

Review the following conceptual topics to learn about the SAP Commerce Cloud (Hybris) Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

**Topics**



> **Note**
>
> There are overall service limits for Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits.

## SAP Commerce Cloud (Hybris) Adapter Capabilities

The SAP Commerce Cloud (Hybris) Adapter enables you to create an integration with an SAP Commerce Cloud application. You can configure the SAP Commerce Cloud (Hybris) Adapter as an invoke connection in an integration in Oracle Integration.

The SAP Commerce Cloud (Hybris) Adapter provides the following benefits:

- Provides invoke (target) support for performing the following types of operations against objects defined under Omni Commerce Connect (OCC) of the SAP Commerce Cloud application:
  - Create or Update (create or update records from SAP Commerce Cloud)
  - Query (get records from SAP Commerce Cloud)
  - Delete (delete records from SAP Commerce Cloud)
- Supports SAP Commerce Cloud APIs protected using OAuth 2.0 two-legged authentication.
- Supports B2B features, if the B2B API AddOn for Omni Commerce Connect (b2boccaddon) is enabled in SAP Commerce Cloud.
- Supports uploading an image in SAP Commerce Cloud.
- Supports implementing secure egress (dedicated NAT gateway) to establish a connection by using a private endpoint. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Configure the Endpoint Access Type.

The SAP Commerce Cloud (Hybris) Adapter is one of many predefined adapters included with Oracle Integration. See the Adapters page in the Oracle Help Center.

# What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

# Workflow to Create and Add an SAP Commerce Cloud (Hybris) Adapter Connection to an Integration

Follow a simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration.

This table lists the workflow steps for both adapter tasks and overall integration tasks, and provides links to instructions for each step.

| Step | Description | More Information |
|---|---|---|
| 1 | Decide where to work | • Work in a project (see why working with projects is preferred in *Using Integrations in Oracle Integration 3*).<br>• Work outside a project. |
| 2 | Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator. | Add the SAP Commerce Cloud (Hybris) Adapter Connection to an Integration |
| 3 | Create the integration. When you do this, you add trigger (source) and invoke (target) connections to the integration. | Create Integrations in *Using Integrations in Oracle Integration 3* and Add the SAP Commerce Cloud (Hybris) Adapter Connection to an Integration |
| 4 | Map data between the trigger connection data structure and the invoke connection data structure. | Map Data in *Using Integrations in Oracle Integration 3* |
| 5 | (Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes). | Manage Lookups in *Using Integrations in Oracle Integration 3* |
| 6 | Activate the integration. | Activate Integrations in *Using Integrations in Oracle Integration 3* |
| 7 | Monitor the integration on the dashboard. | Monitor Integrations During Runtime in *Using Integrations in Oracle Integration 3* |
| 8 | Track payload fields in messages during runtime. | Assign Business Identifiers for Tracking Fields in Messages and Track Integration Instances in *Using Integrations in Oracle Integration 3* |
| 9 | Manage errors at the integration level, connection level, or specific integration instance level. | Manage Errors in *Using Integrations in Oracle Integration 3* |

2

# Create an SAP Commerce Cloud (Hybris) Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate. The following topics describe how to define connections.

**Topics**



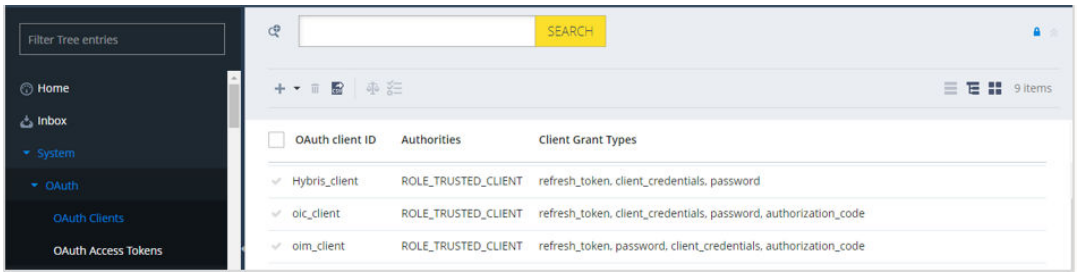
## Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the SAP Commerce Cloud (Hybris) Adapter:

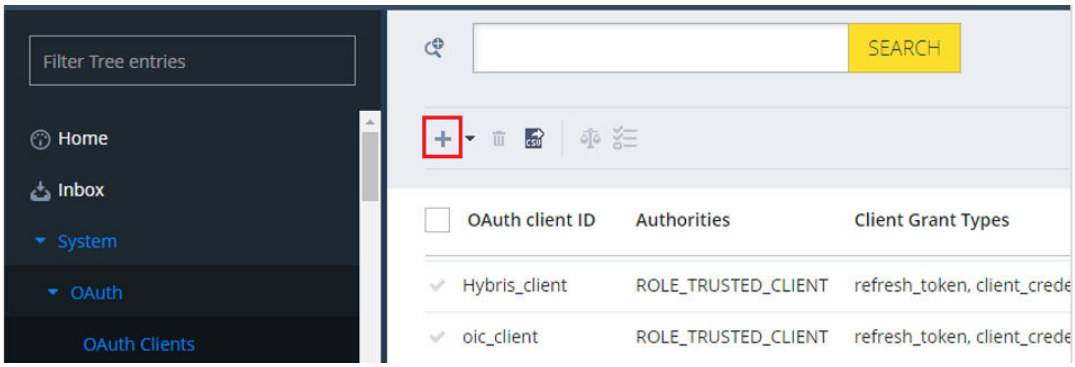
- Create an OAuth Client
- Create and Assign a User to the Customer Manager Group

**Create an OAuth Client**

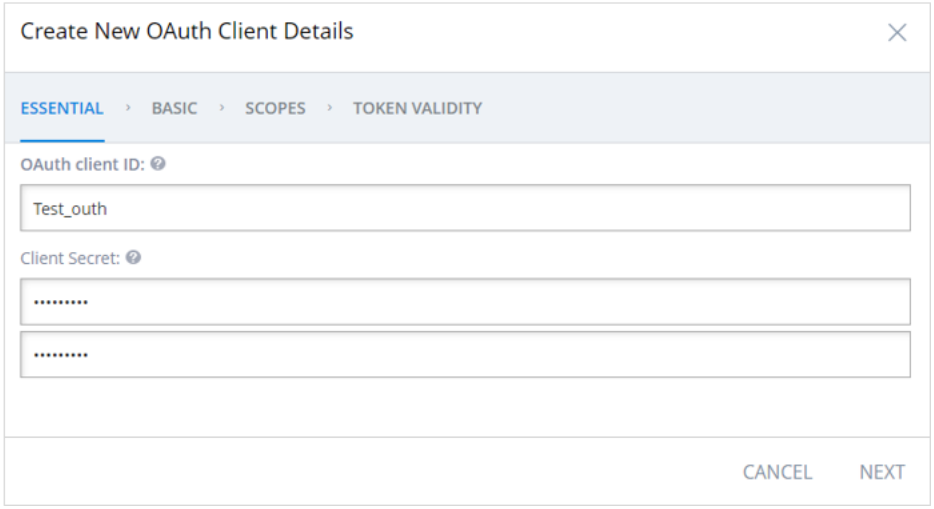
1. Log in to your SAP Commerce Cloud Backoffice and go to **System** > **OAuth** > **OAuth Clients**.




2. Click the add (**+**) icon. The Create New OAuth Client Details dialog box is displayed.

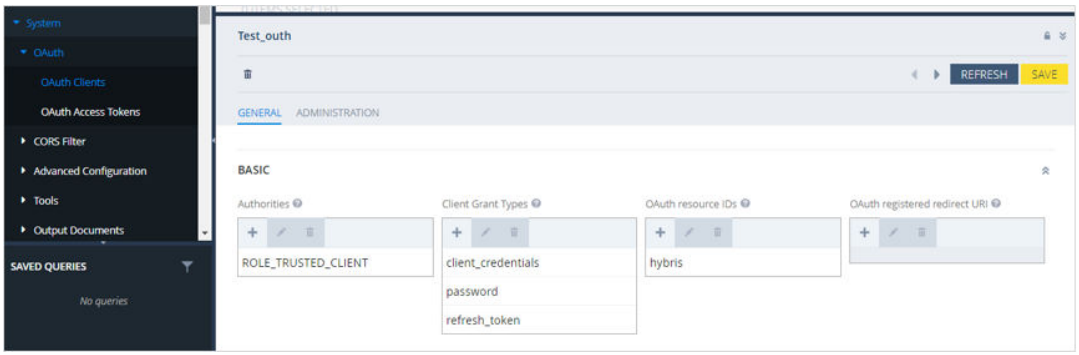


3. Set the OAuth client ID and client secret, and click **NEXT**.




> **Note**
>
> You use the client ID and client secret later when configuring security for your SAP Commerce Cloud (Hybris) Adapter connection. See Configure Connection Security.

4. Enter the following details in the respective fields, and click **NEXT**.

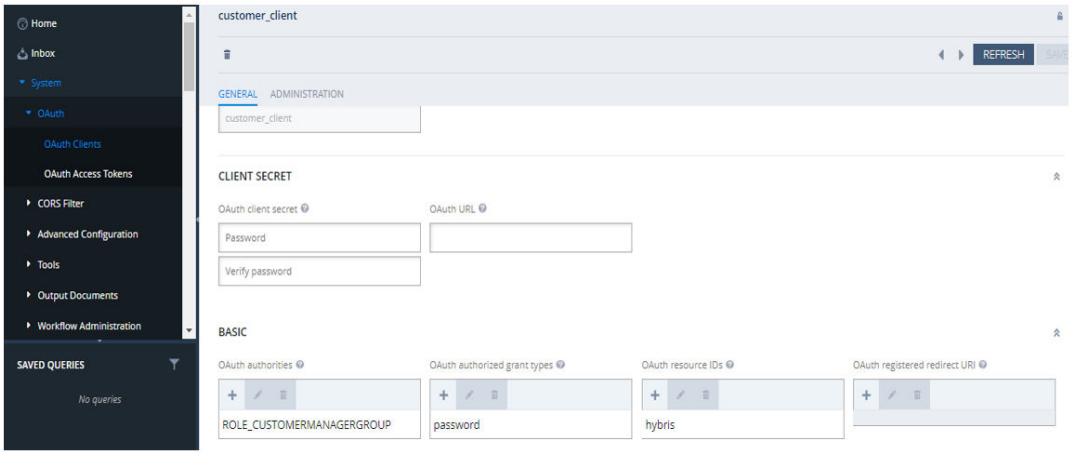



| Element | Description |
|---|---|
| **Authorities** | **ROLE_TRUSTED_CLIENT** |
| **Client Grant Types** | **refresh_token**, **client_credentials**, **password** |
| **OAuth resource IDs** | **hybris** |

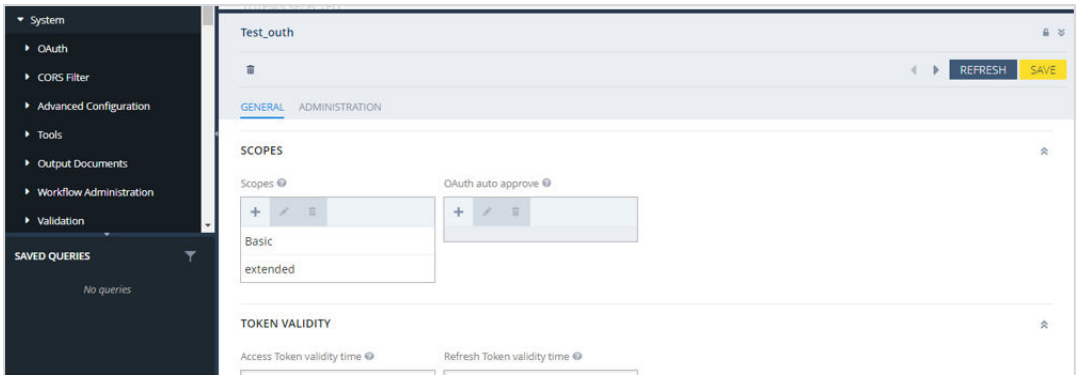
If you need to access Customer Groups services, you must only enter the details described in the following table.

> **Note**
>
> - If you need to access the Customer Groups services, authorization for **ROLE_CUSTOMERMANAGERGROUP** is required.
> - Customer Groups is a secured service and is secured with **ROLE_CUSTOMERMANAGERGROUP**. To access the Customer Groups services, you must be a member of the Customer Manager Group. Therefore, you must assign the Customer Manager Group to the integration user. See Create and Assign a User to the Customer Manager Group.
> - Configure Resource Owner Password Credentials (ROPC) security for the SAP Commerce Cloud (Hybris) Adapter connection to access the customer group services.

| Element | Description |
|---|---|
| **Authorities** | **ROLE_CUSTOMERMANAGERGROUP** |
| **Client Grant Types** | **password** |
| **OAuth resource IDs** | **hybris** |




5. Enter `basic` and `extended` in the **Scopes** field, and click **NEXT**.




6. In the **Access Token validity time** field, enter the required time (in seconds) that depends on the frequency of integration execution (for example, `28800`). In the **Refresh Token**

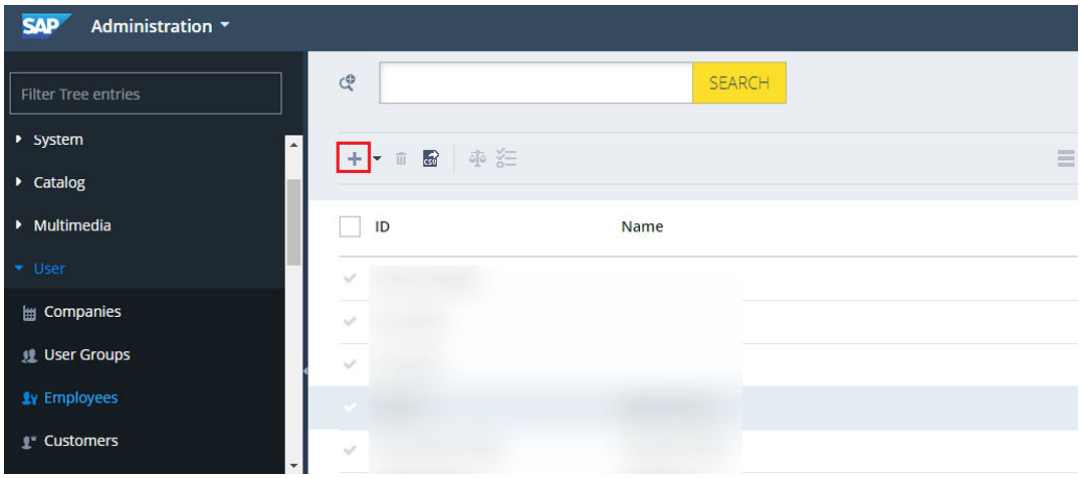
**validity time** field, enter a value greater than Access Token validity time (for example, `32400`).

7. Click **DONE**.

### Create and Assign a User to the Customer Manager Group

Perform the following steps to create and assign a user to the Customer Manager Group.

1. Log in to your SAP Commerce Cloud Backoffice and go to **User** > **Employees**.
2. Click the add (**+**) icon. The Create New Employee dialog box is displayed.

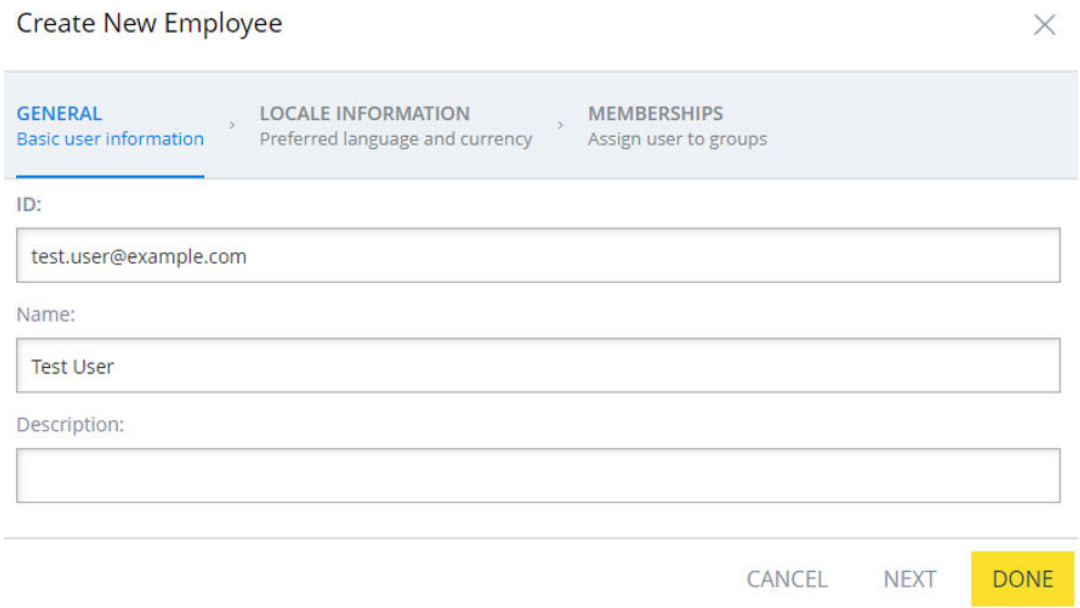



3. Enter the ID in the following format:

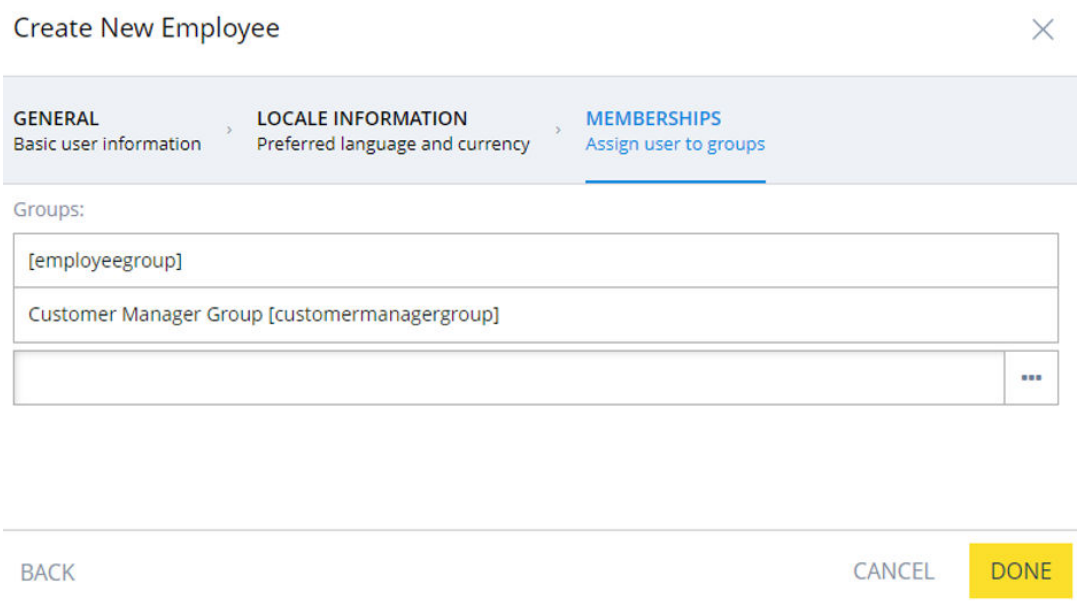
   `firstname.lastname@domain.com`

   For example: `c.smith@example.com`

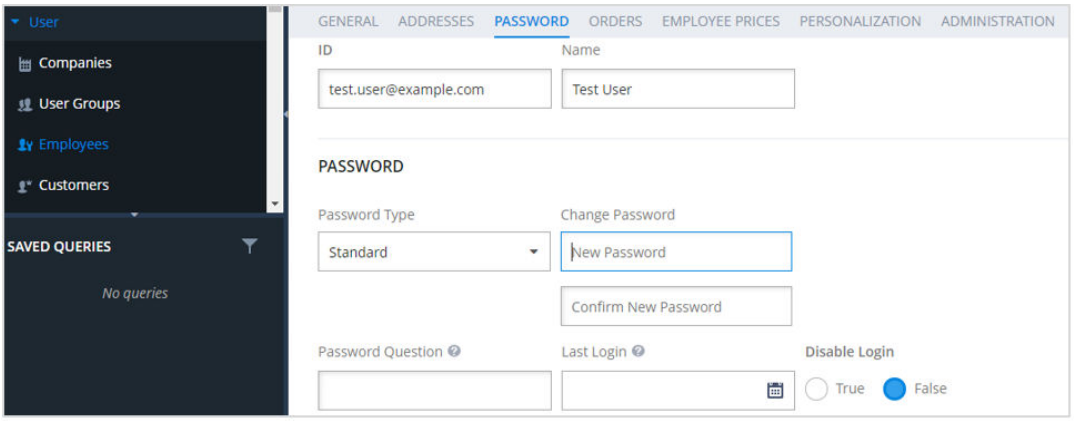
4. Enter the additional required details in the respective fields, and click **NEXT**.



5. Under **MEMBERSHIPS**, search for **customermanagergroup**, assign the user to the Customer Manager Group, and click **DONE**.




6. Once the user is created and assigned to Customer Manager Group, click **Employees** under **User** on the home page.
7. Click the respective customer ID, and click the **PASSWORD** tab.




8. Enter a new password for the customer user, re-enter the password for confirmation, and click **SAVE**.

# Common Customization Scenarios

You must perform specific annotations to ensure that SAP Commerce Cloud reflects customizations in the SAP Commerce Cloud (Hybris) Adapter.

Three main types of customizations are possible in the Omni Commerce Connect (OCC) services of SAP Commerce Cloud. To make customizations available in the SAP Commerce Cloud (Hybris) Adapter, follow these guidelines:

- Add New Custom Fields in the Standard Operation/API

- Add a New Custom API/Operation Under a Standard Object
- Add a New Custom Object

### Add New Custom Fields in the Standard Operation/API

Ensure that the custom fields are annotated with `@ApiParam` so that the fields can be discovered in the Swagger metadata document.

```
@ApiParam(value = "Sorting method applied to the return results.") @RequestParam(required = false) final String sort,
@ApiParam(value = "The context to be used in the search query.") @RequestParam(required = false) final String searchQueryContext,
@ApiFieldsParam @RequestParam(defaultValue = DEFAULT_FIELD_SET) final String fields,
@ApiParam(value = "Searched Based on Product Name", required = false) @RequestParam(required = false) final String productName,
final HttpServletResponse response){
```

### Add a New Custom API/Operation Under a Standard Object

Ensure that the operation (method) is annotated with `@ApiOperation` and `@ApiBaseSiteIdParam` so that the operation can be discovered in the Swagger metadata document.

```
@RequestMapping(value = "/getStoreByName/{storeName}", method = RequestMethod.GET)
@ApiOperation(nickname = "getStoreByName", value = "Get a store by name", notes = "Returns store based on its unique name.")
@ApiBaseSiteIdParam
@ResponseBody
public PointOfServiceWsDTO getStoreByName(@ApiParam(value = "Store identifier (by store name)", required = true)
```

### Add a New Custom Object

Ensure that the new object (controller class) is annotated with `@Api(tags = “<<Object Name>>”)` so that the object can be discovered in the Swagger metadata document. You can follow the above-mentioned information to add the operations and fields.

```
/**
package com.ocicustomoccwebservice.v2.controller;

import de.hybris.platform.webservicescommons.cache.CacheControl;


/**
 * @author Anand.Mund
 *
 */

@Controller
@RequestMapping(value = "/{baseSiteId}/brand")
@CacheControl(directive = CacheControlDirective.PRIVATE)
@Api(tags = "Brand")
public class BrandController extends BaseController{
   private static final Logger LOG = LoggerFactory.getLogger(BrandController.class);

   @Resource(name = "brandFacade")
   private BrandFacade brandFacade;

   @RequestMapping(value ={ "/search" }, method = RequestMethod.GET)
   @ResponseBody
   @ApiOperation(nickname = "getBrand", value = "Get the list of brand for a particular basestore..")
   @ApiBaseSiteIdParam
   public BrandWsDTO getBrand(@ApiFieldsParam @RequestParam(defaultValue = DEFAULT_FIELD_SET) final String fields)   {
      //final BrandWsDTO brandDto = new BrandWsDTO();
      final BrandData brandData = brandFacade.getBrand();

      return getDataMapper().map(brandData, BrandWsDTO.class, fields);
   }

}
```

> **Note**
>
> Replace `<<Object Name>>` with the actual name of the object that you need to add.

## Upload an Image

You can upload an image for a product in SAP Commerce Cloud using customizations in the Omni Commerce Connect (OCC) services of SAP Commerce Cloud.

Perform the following steps to upload an image for a product:

1. Add a new object in the OCC services of SAP Commerce Cloud. See Add a New Custom Object.

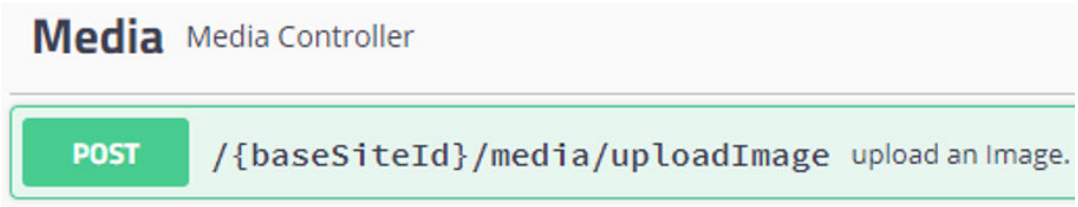
```
 * @author Anand.Mund
 *
 */

@Controller
@Api(tags = "Media")
@RequestMapping(value = "/{baseSiteId}/media")
@CacheControl(directive = CacheControlDirective.PRIVATE)
public class MediaController extends BaseController{
   private static final Logger LOGGER = LoggerFactory.getLogger(MediaController.class);

   @Resource
   private MediaFacade mediaFacade;
   @Resource
   private DataMapper dataMapper;
   @Resource
   private CustomProductFacade oicProductFacade;

   @RequestMapping(value = "/uploadImage", method = RequestMethod.POST/*, consumes = MediaType.MULTIPART_FORM_DATA_VALUE*/)
   @ResponseStatus(value = HttpStatus.CREATED)
   @ResponseBody
   @ApiOperation(nickname = "uploadImage", value = "upload an Invoice")
   @ApiBaseSiteIdParam
   public CustomMediaWsDTO uploadMediaImage(
         @ApiParam(value = "The MediaData containing the data for the associated media item to be created.", required = true)
         @ModelAttribute("media") final MediaData media,
         @ApiParam(value = "The unique identifier of the product to which the new image will be linked.", required = true)
         @RequestParam("productId") final String productId,
         @ApiParam(value = "The file representing the actual binary contents of the media to be created.", required = true)
         @RequestParam("file_field_1") final MultipartFile multiPart,
         final HttpServletRequest httpRequest, final HttpServletResponse httpResponse) throws IOException {
```

2. Once the custom object is created, open the Swagger file. The custom object (that is, the media and service (endpoint)), is displayed in the Swagger file.



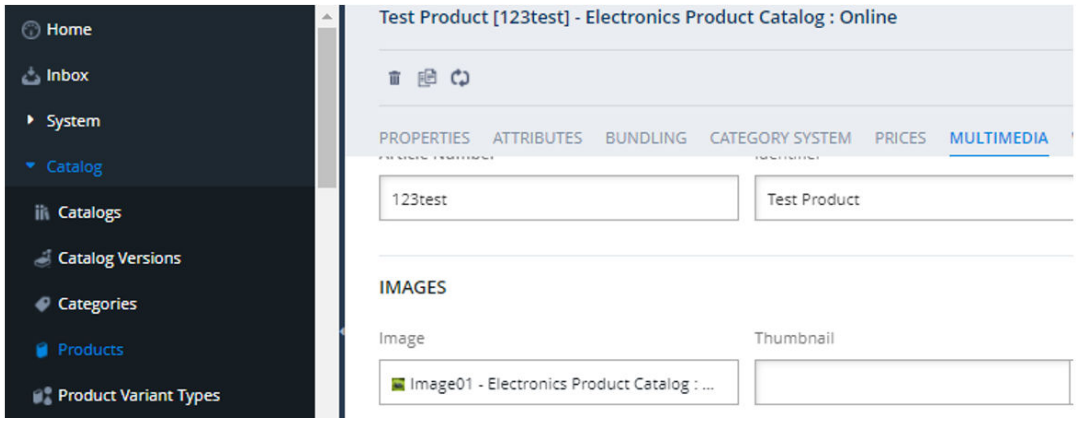

   You can send an image in Base64 format to Oracle Integration using a SOAP Adapter connection.

3. Configure the SOAP Adapter as a trigger connection and the SAP Commerce Cloud Adapter as an invoke connection in an integration in Oracle Integration.
4. In the mapper, perform the mapping. The image is decoded to the attachment reference type using functions in the mapper. The same image is uploaded for the product based on the product ID passed in the request payload.

To verify the uploaded image in Backoffice:

1. Log in to your SAP Commerce Cloud Backoffice application.

2. Navigate to **Catalog** > **Product**.
3. Select the respective product and click the **MULTIMEDIA** tab.




4. Click the uploaded image. The image is displayed.

# Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

> **Note**
>
> You can also create a connection in the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Create connection panel opens.
2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

| Element | Description |
|---|---|
| **Name** | Enter a meaningful name to help others find your connection when they begin to create their own integrations. |
| **Identifier** | Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES_OPPORTUNITY`). |
| **Role** | Select the role (direction) in which to use this connection.<br>**Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations.<br>When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select.<br>For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error. |
| **Keywords** | Enter optional keywords (tags). You can search on the connection keywords on the Connections page. |
| **Description** | Enter an optional description of the connection. |
| **Share with other projects** | **Note**: This field only appears if you are creating a connection in a project.<br>Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects.<br>When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources.<br>See Add and Share a Connection Across a Project. |

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

# Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Properties** section.

   The Connection Properties dialog is displayed.

2. In the **SAP Commerce Cloud Instance URL** field, enter your SAP Commerce Cloud instance URL.

# Configure Connection Security

Configure security for your SAP Commerce Cloud (Hybris) Adapter connection.

1. Go to the **Security** section.
2. From the **Security Policy** list, select the security policy.
   - **Client Credentials**
   - **Resource Owner Password Credentials**
3. If you select **Client Credentials Policy**.
   a. In the **Client Id** field, enter the client ID that you obtained after performing the steps in the prerequisites section. See Create an OAuth Client.
   b. In the **Client Secret** field, enter the client secret that you obtained after performing the steps in the prerequisites section. See Create an OAuth Client.
4. If you select **Resource Owner Password Credentials**.
   a. In the **Client Id** field, enter the client ID that you obtained after performing the steps in the prerequisites section. See Create an OAuth Client.
   b. In the **Client Secret** field, enter the client secret that you obtained after performing the steps in the prerequisites section. See Create an OAuth Client.
   c. In the **Username** field, enter the username.
   d. In the **Password** field, enter the password.

# Configure the Endpoint Access Type

Configure access to your endpoint. Depending on the capabilities of the adapter you are configuring, options may appear to configure access to the public internet, to a private endpoint, or to an on-premises service hosted behind a fire wall.

- Select the Endpoint Access Type
- Ensure Private Endpoint Configuration is Successful

### Select the Endpoint Access Type

1. Go to the **Access type** section.
2. Select the option for accessing your endpoint.

| Option | This Option Appears If Your Adapter Supports ... |
|---|---|
| **Public gateway** | Connections to endpoints using the public internet. |
| **Private endpoint** | Connections to endpoints using a private virtual cloud network (VCN).<br>**Note**: To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Troubleshoot Private Endpoints in *Using Integrations in Oracle Integration 3*. |

### Ensure Private Endpoint Configuration is Successful

- To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- When configuring an adapter on the Connections page to connect to endpoints using a private network, specify the fully-qualified domain name (FQDN) and *not* the IP address. If you enter an IP address, validation fails when you click **Test**.

## Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your adapter connection uses a Web Services Description Language (WSDL) file. Only some adapter connections use WSDLs.

| If Your Connection... | Then... |
|---|---|
| Doesn't use a WSDL | The test starts automatically and validates the inputs you provided for the connection. |
| Uses a WSDL | A dialog prompts you to select the type of connection testing to perform:<br>• **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL.<br>• **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL. |

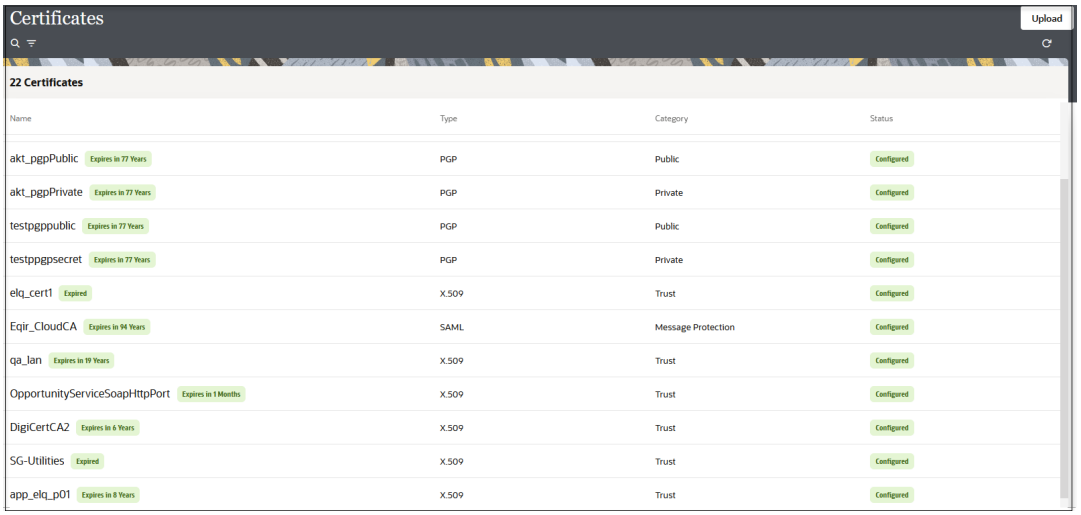
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection is successful.
3. When complete, click **Save**.

# Upload a Certificate to Connect with External Services

Certificates allow Oracle Integration to connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then import it into Oracle Integration.

If you make an SSL connection in which the root certificate does not exist in Oracle Integration, an exception error is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration.

1. Sign in to Oracle Integration.
2. In the navigation pane, click **Settings**, then **Certificates**.
   All certificates currently uploaded to the trust store are displayed on the Certificates page.
3. Click **Filter** to filter by certificate expiration date, status, and type. Certificates installed by the system cannot be deleted.




4. Click **Upload** at the top of the page.
   The Upload certificate panel is displayed.
5. Enter an alias name and optional description.
6. In the **Type** field, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
   - Digital Signature
   - X.509 (SSL transport)
   - SAML (Authentication & Authorization)
   - PGP (Encryption & Decryption)
   - Signing key

### Digital Signature

The digital signature security type is typically used with adapters created with the Rapid Adapter Builder. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

1. Click **Browse** to select the digital certificate. The certificate must be an X509Certificate. This certificate provides inbound RSA signature validation. See RSA Signature Validation in *Using the Rapid Adapter Builder with Oracle Integration 3*.
2. Click **Upload**.

### X.509 (SSL transport)

1. Select a certificate category.
   a. **Trust**: Use this option to upload a trust certificate.
      i. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.
   b. **Identity**: Use this option to upload a certificate for two-way SSL communication.
      i. Click **Browse**, then select the keystore file (`.jks`) to upload.
      ii. Enter the comma-separated list of passwords corresponding to key aliases.

      > **Note**
      >
      > When an identity certificate file (`.jks`) contains more than one private key, all the private keys must have the same password. If the private keys are protected with different passwords, the private keys cannot be extracted from the keystore.

      iii. Enter the password of the keystore being imported.
   c. Click **Upload**.

### SAML (Authentication & Authorization)

1. Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.
2. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.
3. Click **Upload**.

### PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.
   a. **Private**: Uses a private key of the target location to decrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. Enter the PGP private key password.
   b. **Public**: Uses a public key of the target location to encrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**.
         - **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted

messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content.

- **No** causes the message to be sent in binary format.

iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. The following supported cipher algorithms are FIPS-compliant:

- AES128
- AES192
- AES256
- TDES

c. Click **Upload**.

### Signing key

A signing key is a secret key used to establish trust between applications. Signing keys are used to sign ID tokens, access tokens, SAML assertions, and more. Using a private signing key, the token is digitally signed and the server verifies the authenticity of the token by using a public signing key. You must upload a signing key to use the OAuth Client Credentials using JWT Client Assertion and OAuth using JWT User Assertion security policies in REST Adapter invoke connections. Only PKCS1- and PKCS8-formatted files are supported.

1. Select **Public** or **Private**.
2. Click **Browse** to upload a key file.
   If you selected **Private**, and the private key is encrypted, a field for entering the private signing key password is displayed after key upload is complete.
3. Enter the private signing key password. If the private signing key is not encrypted, you are not required to enter a password.
4. Click **Upload**.

3

# Add the SAP Commerce Cloud (Hybris) Adapter Connection to an Integration

When you drag the SAP Commerce Cloud (Hybris) Adapter into the invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the SAP Commerce Cloud (Hybris) Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the SAP Commerce Cloud (Hybris) Adapter as an invoke in an integration. The SAP Commerce Cloud (Hybris) Adapter cannot be used as a trigger in an integration.

**Topics**



## Basic Info Page

You can enter a name and description on the Basic Info page of each adapter in your integration.

| Element | Description |
|---|---|
| **What do you want to call your endpoint?** | Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters:<br>• No blank spaces (for example, `My Inbound Connection`)<br>• No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens<br>• No multibyte characters |
| **What does this endpoint do?** | Enter an optional description of the connection's responsibilities. For example:<br>`This connection receives an inbound request to synchronize account information with the cloud application.` |

# Invoke Actions Page

Select the action to perform on SAP Commerce Cloud.

| Element | Description |
|---|---|
| **Which action do you want to perform on SAP Commerce Cloud?** | • **Query**: Retrieves information from the SAP Commerce Cloud application corresponding to the selected object and operation.<br>• **Create or Update**: Performs operations for user creation, update payment details, and so on.<br>• **Delete**: Deletes records from the SAP Commerce Cloud application. |

# Invoke Operations Page

Select the object and operation to perform on the object.

| Element | Description |
|---|---|
| **Select Object** | Use the scrolling list to select an object within the selected action. |
| **Filter by Object Name** | Type the initial letters of the object name to filter the display of names in the list. |
| **Select Operation** | Select an operation name, such as **Get Named Accounts**. |
| **Filter by Operation Name** | Type the initial letters of the operation name to filter the display of names in the list. |

# Summary Page

You can review the specified adapter configuration values on the Summary page.

| Element | Description |
|---|---|
| **Summary** | Displays a summary of the configuration values you defined on previous pages of the wizard.<br>The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file.<br>To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**.<br>To cancel your configuration details, click **Cancel**. |

4

# Implement Common Patterns Using the SAP Commerce Cloud (Hybris) Adapter

You can use the SAP Commerce Cloud (Hybris) Adapter to implement the following common pattern.

**Topics:**



## Synchronize a Salesforce User with a Customer in the SAP Commerce Cloud Application

This use case provides an overview of how to synchronize a Salesforce user with a customer in the SAP Commerce Cloud application.

1. Create SAP Commerce Cloud (Hybris) Adapter and Salesforce Adapter connections.
2. Create an app-driven orchestrated integration.
3. Drag the Salesforce Adapter connection into the integration.
4. Configure the workflow rule and outbound messages in the Salesforce application. The same WSDL must be uploaded in the integration.
5. Drag and drop the SAP Commerce Cloud (Hybris) Adapter into the integration.
6. Configure the SAP Commerce Cloud endpoint:
    a. On the Basic info page, provide an endpoint name, and click **Next**.
    b. On the Action page, select **Create or Update**, and click **Next**.
    c. On the Operations page, select **Users** as an object, select **Create User** as an operation, and click **Next**.
    d. On the Summary page, review your selections, and click **Done**.
7. In the mapper, map the **First Name**, **Last Name**, **Email**, and **Title** elements to the respective fields of SAP Commerce Cloud.
8. Provide the value for **baseSiteId** and the password (password must include at least one uppercase, lowercase, number and special character) in the mapper. The completed integration looks as follows.
9. Specify the tracking variable, save, and close the integration.
10. Activate the integration.
11. Create the user in Salesforce. The same user now reflects on the SAP Commerce Cloud application.
12. To verify the created user, log in to your SAP Commerce Cloud account, and go to **User** > **Customers**.