Oracle® Cloud

Using the GitHub Adapter with Oracle Integration 3

F86026-05
October 2024

Oracle Cloud Using the GitHub Adapter with Oracle Integration 3,

F86026-05

Primary Author: Oracle Corporation

Contents

Preface

Preface

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

> **Note:**
>
> The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Resources
- Conventions

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 `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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 GitHub Adapter

Review the following topics to learn about the GitHub 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:

- GitHub Adapter Capabilities
- GitHub Adapter Restrictions
- What Application Version Is Supported?
- Workflow to Create and Add a GitHub Adapter Connection to an Integration

GitHub Adapter Capabilities

The GitHub Adapter enables you to create an integration in Oracle Integration that connects with GitHub. You can configure the GitHub Adapter as a trigger connection or an invoke connection in an integration in Oracle Integration.

The GitHub Adapter supports invoke connections with the following capabilities:

- Provides support for performing various operations such as Create, Update, Get, List, Delete, Compare, Search, Merge, and others on the selected business object.
- Supports OAuth 2.0 Authorization Code Credentials authentication for invoke connections.

The GitHub Adapter supports trigger connections with the following capabilities:

- Provides support for subscribing to Repository and Organization events (Organization Credentials only).
- Authenticates trigger connections using composite security policies: HMAC (Hash-based Message Authentication Code) Signature Validation for incoming requests and OAuth 2.0 Authorization Code Credentials for outgoing calls.

The GitHub Adapter is one of many predefined adapters included with Oracle Integration.

GitHub Adapter Restrictions

Note the following GitHub Adapter restriction.

If an OAuth application generates more than 10 tokens for a single user with identical scopes, the oldest tokens associated with the same user, application, and scope combination are automatically revoked. This limitation applies exclusively to OAuth applications. See Token revoked due to excess of tokens for an OAuth app with the same scope.

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 a GitHub Adapter Connection to an Integration

You follow a very 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	Access Oracle Integration.	Go to `https://`*`instance_URL`*`/ic/home`
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.	Create a GitHub Adapter Connection
3	Create the integration. When you do this, you add trigger (source) and invoke (target) connections to the integration.	Understand Integration Creation and Best Practices in *Using Integrations in Oracle Integration 3* and Add the GitHub 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 an Integration 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 a GitHub Adapter Connection

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

Topics:

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the GitHub Adapter.

- Register a GitHub Application
- Authorize a GitHub Application
- Create an OAuth Application
- Create Client Identifier
- Configure the Client Application
- Add Roles to the Client Application

Register a GitHub Application

You have the option to register a GitHub application either under your personal account or an organization for which you have been granted permission to manage all owned applications. See Registering a GitHub App.

- During the registration process, enter the callback URL of your application in the following format:

  ```
  https://OIC_instance_URL/icsapis/agent/oauth/callback
  ```

- Under **Permissions**, choose the permissions that your application needs. You need to grant Repository Permissions, Organization Permissions, and Account Permissions. For each permission, select the drop-down menu and click **Read-only**, **Read & write**, or **No access**.
- Click **Generate a New Client Secret**.

> **Note:**
>
> Copy the client ID and client secret. You'll need to enter those values on the Connections page when you configure security for your GitHub Adapter connection in Oracle Integration. See Configure Connection Security.

Authorize a GitHub Application

When you authorize a GitHub application, you are providing the application with access to your GitHub account in accordance with the permissions it requested during the authorization process.

1. Click your organization name.
2. Click **Install**.
3. Select either **All repositories** or **Only select repositories**.
4. Click **Authorize & Request**.

Create an OAuth Application

Create and register an OAuth application under your personal account or under any organization. See Create an OAuth app.

- During the registration process, enter the callback URL of your application in the following format:

  ```
  https://OIC_instance_URL/icsapis/agent/oauth/callback
  ```

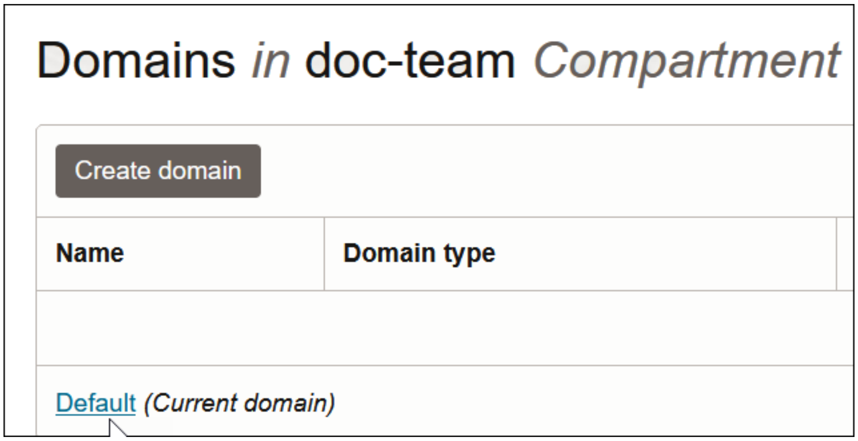
- During the registration process, provide the Webhook URL, specifying the destination where you want to receive notifications.

> **Note:**
>
> - OAuth applications can have multiple Webhook URLs.
> - Copy the client ID and client secret. You'll need to enter those values on the Connections page when you configure security for your GitHub Adapter connection in Oracle Integration. See Configure Connection Security.

Create Client Identifier

Access the Identity Domain

1. Log in to the Oracle Cloud Infrastructure Console with your identity domain administrator credentials.
   a. In the navigation pane, click **Identity & Security**.
   b. Click **Domains**.
   c. Select your compartment.
   d. Click the identity domain.



Domains *in* doc-team *Compartment*

Name	Domain type
Default *(Current domain)*	

e. In the navigation pane, click **Integrated applications**. This is the location at which you create the client application for your grant type.

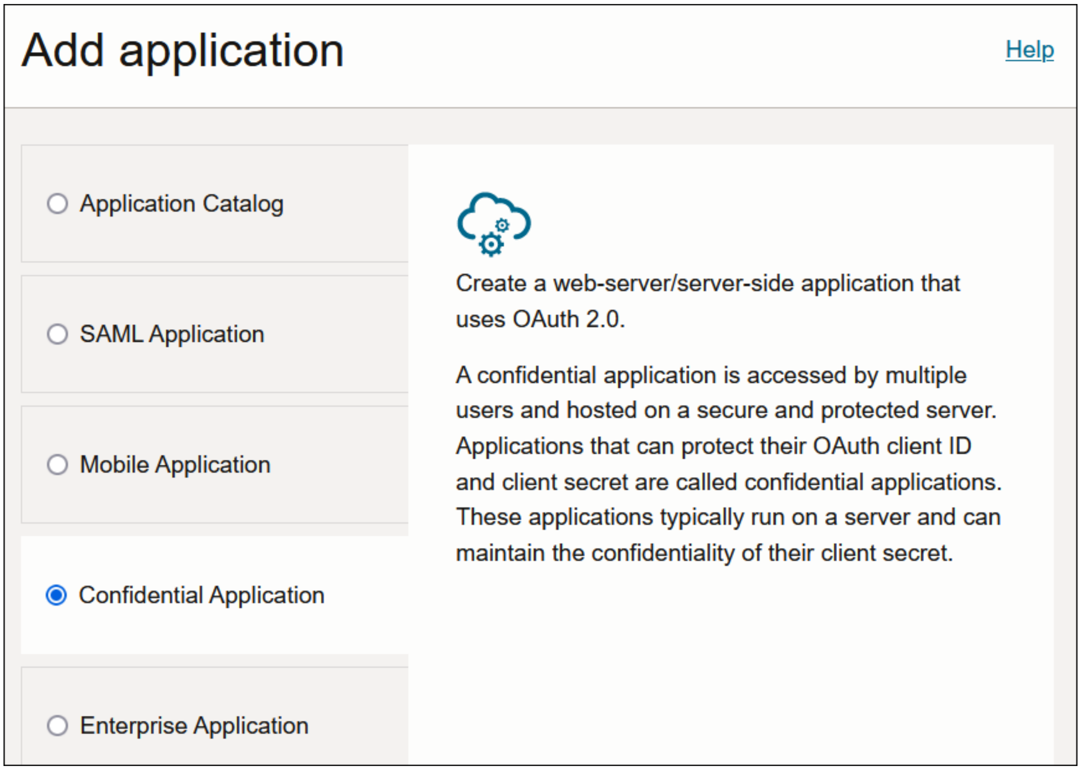
Prerequisites for Resource Owner Password Credentials

To trigger the integration with OAuth, a client application is required.

- Configure the Client Application
- Add Roles to the Client Application

Configure the Client Application

1. Click **Add application**.
2. Select **Confidential Application**, then click **Launch workflow**.



Add application

○ Application Catalog

○ SAML Application

○ Mobile Application

● Confidential Application

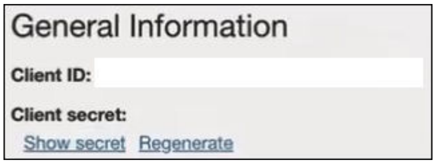
○ Enterprise Application

Create a web-server/server-side application that uses OAuth 2.0.

A confidential application is accessed by multiple users and hosted on a secure and protected server. Applications that can protect their OAuth client ID and client secret are called confidential applications. These applications typically run on a server and can maintain the confidentiality of their client secret.

3. Enter a name. The remaining fields on this page are optional and can be ignored.

4. Click **Next**.
5. Select **Skip for later**.
6. Click **Activate**, and then **Activate application** to activate the client application for use.
7. In the **General Information** section, note the client ID value. This value is required for configuring your connection to establish identity.



General Information

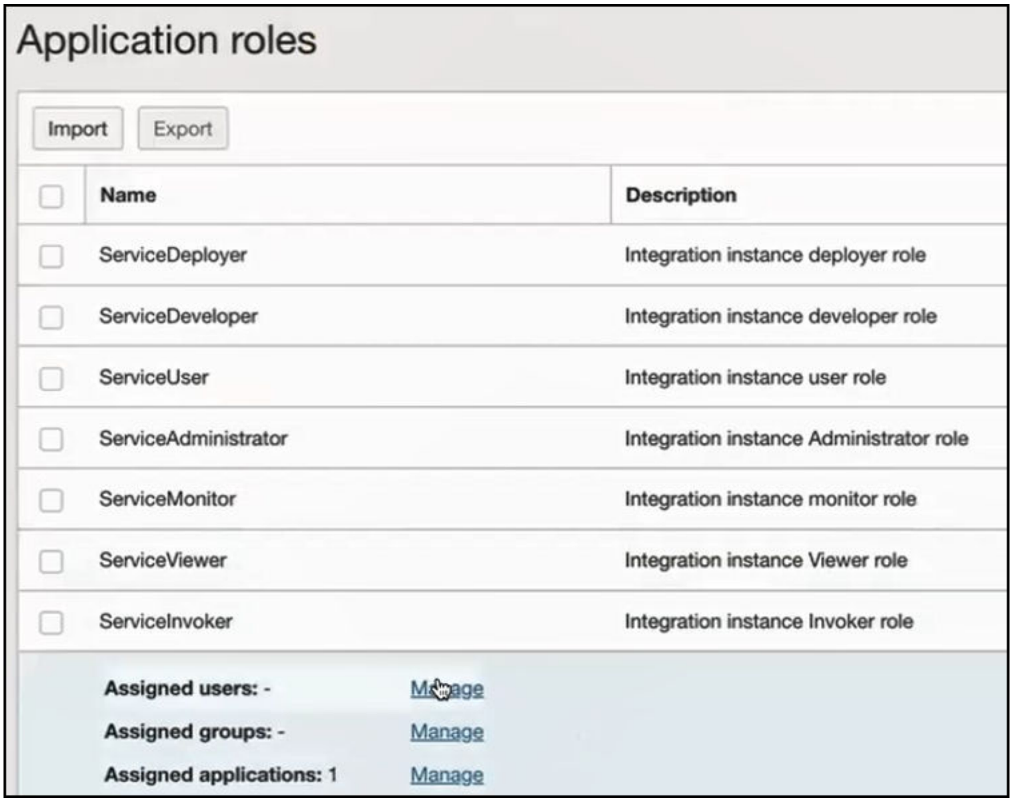
Client ID:

Client secret:

> **Note:**
>
> Copy the client ID and client secret. You'll need to enter those values on the Connections page when you configure security for your GitHub Adapter connection in Oracle Integration. See Configure Connection Security.

Add Roles to the Client Application

1. In the navigation pane, click **Oracle Cloud Services**.
2. Select the specific application corresponding to the Oracle Integration instance.
3. In the navigation pane, click **Application roles**.
4. Expand **ServiceInvoker**, then click **Manage** next to either **Assigned users** or **Assigned groups**. For example, if you click **Assigned users**:



□	Name	Description
□	ServiceDeployer	Integration instance deployer role
□	ServiceDeveloper	Integration instance developer role
□	ServiceUser	Integration instance user role
□	ServiceAdministrator	Integration instance Administrator role
□	ServiceMonitor	Integration instance monitor role
□	ServiceViewer	Integration instance Viewer role
□	ServiceInvoker	Integration instance Invoker role

Assigned users: -	Manage
Assigned groups: -	Manage
Assigned applications: 1	Manage

5. Click **Show available users**.
6. Select the user and click **Assign**, then click **Close**.

Create a Connection

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

To create a connection in Oracle Integration:

1. In the navigation pane, click **Design**, then **Connections**.
2. Click **Create**.

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

3. In the Create connection panel, 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.
4. 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.

Element	Description
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 (trigger, invoke, or both). Only the roles supported by the adapter are displayed for selection. 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. 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. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. 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. See Add and Share a Connection Across a Project.

5. Click **Create**.

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

Configure Connection Security

Configure security for your GitHub Adapter connection.

1. Go to the **Security** section.
2. For trigger connections:

   Composite Security Policies is displayed. This value cannot be changed.

   a. In the **Client Id** field, enter the client ID that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

b. In the **Client Secret** field, enter the client secret that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

c. In the **Shared Secret** field, enter the shared secret that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

d. In the **Client Identifier** field, enter the client ID that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection

e. (Required only for OAuth applications) In the **Scope** field, enter the required scopes. See Scopes for OAuth apps.

> **Note:**
>
> When creating and installing a GitHub application, the scope is defined during the setup process. Conversely, for an OAuth application, you input the necessary scope value on the Connections page in Oracle Integration.

f. Click **Provide Consent** to verify the connection properties. The GitHub login page is displayed.

g. Enter your GitHub login credentials.
Once you see an `Authenticated!!` message, you can test your connection.

3. For invoke connections:

   OAuth 2.0 Authorization Code Credentials is displayed. This value cannot be changed.

a. In the **Client Id** field, enter the client ID that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

b. In the **Client Secret** field, enter the client secret that you obtained after performing the prerequisite steps. See Prerequisites for Creating a Connection.

c. (Required only for OAuth applications) In the **Scope** field, enter the required scopes. See Scopes for OAuth apps.

> **Note:**
>
> When creating and installing a GitHub application, the scope is defined during the setup process. Conversely, for an OAuth application, you input the necessary scope value on the Connections page in Oracle Integration.

d. Click **Provide Consent** to verify the connection properties.
The GitHub login page is displayed.

e. Enter your GitHub login credentials.
Once you see an `Authenticated!!` message, you can test your connection.

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: • **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. • **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.

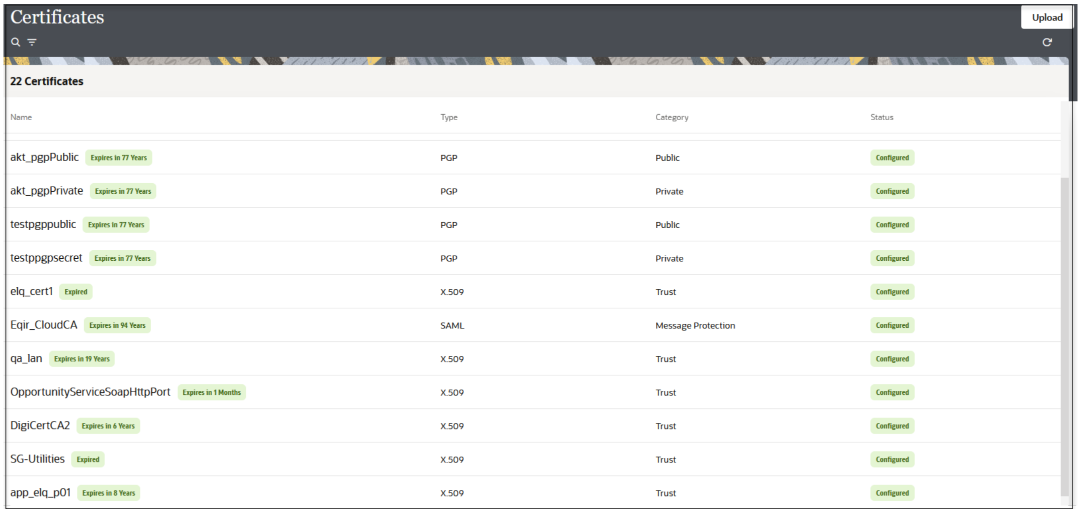
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection 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 name, certificate expiration date, status, type, category, and installation method (user-installed or system-installed). Certificates installed by the system cannot be deleted.



22 Certificates

Name	Type	Category	Status
akt_pgpPublic Expires in 77 Years	PGP	Public	Configured
akt_pgpPrivate Expires in 77 Years	PGP	Private	Configured
testpgppublic Expires in 77 Years	PGP	Public	Configured
testppgpsecret Expires in 77 Years	PGP	Private	Configured
elq_cert1 Expired	X.509	Trust	Configured
Eqir_CloudCA Expires in 94 Years	SAML	Message Protection	Configured
qa_lan Expires in 19 Years	X.509	Trust	Configured
OpportunityServiceSoapHttpPort Expires in 1 Months	X.509	Trust	Configured
DigiCertCA2 Expires in 6 Years	X.509	Trust	Configured
SG-Utilities Expired	X.509	Trust	Configured
app_elq_p01 Expires in 8 Years	X.509	Trust	Configured

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**.

Refresh Integration Metadata

You can manually refresh the currently-cached metadata available to adapters that have implemented metadata caching.

Metadata changes typically relate to customizations of integrations, such as adding custom objects and attributes to integrations. There may also be cases in which integrations have been patched, which results in additional custom objects and attributes being added. This option is similar to clearing the cache in your browser. Without a manual refresh, a staleness check is only performed when you drag a connection into an integration. This is typically sufficient, but in some cases you may know that a refresh is required. For these cases, the **Refresh Metadata** menu option is provided.

> **Note:**
>
> The **Refresh Metadata** menu option is only available with adapters that have implemented metadata caching.

1. In the navigation pane, click **Design**, then **Connections**.
2. Hover over the connection to refresh.
3. Click **Actions** ⋯, then select **Refresh metadata**.

   A message is displayed indicating that the refresh was successful.

3

Add the GitHub Adapter Connection to an Integration

When you drag the GitHub Adapter into the trigger or invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the GitHub Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the GitHub Adapter as a trigger or invoke in an integration.

Topics:

Trigger Basic Info Page

Specify a name and description, select a business object, and select a trigger operation on the Basic Info page for the trigger connection 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. For example, if you are creating an invoke connection to delete a blob, you may want to name it `InvokeDelBlobs`. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection’s responsibilities.
Select Business Object	Select a business object: • **Repository Events** • **Organization Event (Org Credentials Only)**
Trigger	Select an operation name to perform such as **Subscribe to Repository Event** or **Subscribe to Organization Event**.

Trigger Configuration Page

Select a repository and event name.

If you select **Repository Events** on the Basic Info page, the following options are displayed.

Element	Description
Repository	Select the required repository from the drop-down list.
Event Name	Select an event from the drop-down list such as **Repository Event**, **Pull Request Event**, **Commit Comment Event**, and so on.

If you select **Organization Event (Org Credentials Only)** on the Basic Info page, the following options are displayed.

> **Note:**
>
> When you select the **Organization Event (Org Credentials Only)** option on the Basic Info page, the connection details on the Connection page should pertain to the organization. If not, an empty drop-down list for the organization name field is displayed. If you attempt to select from this empty drop-down list, an error message is displayed.

Element	Description
Organization	Select the organization.
Event Name	Select an event from the drop-down list, such as **Git Push Events**, **Issues Event**, **Release Event**, and so on.

Invoke Basic Info Page

Specify a name and description, select a business object, and select an operation on the Basic Info page for each invoke connection 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. For example, if you are creating an invoke connection to delete a blob, you may want to name it `InvokeDelBlobs`. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection’s responsibilities.

Element	Description
Select Business Object	Select a business object: • **Branch** • **Users** • **Issues** • **Commits** • **Releases** • **Search** • **Forks** • **Repository** • **Organization** • **Deploy Keys** • **Teams** • **Pulls**
Action	Select an operation to perform such as **Create a release**, **Get a repository**, **Delete an organization**, **Update the authenticated user**, **Search Users**, and so on.

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. 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. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

4

Implement Common Patterns Using the GitHub Adapter

You can use the GitHub Adapter to implement the following common pattern.

Topics:

> **Note:**
>
> Oracle Integration offers a number of prebuilt integrations, known as *recipes*, that provide you with a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements. Depending upon the solution provided, a variety of adapters are configured in the prebuilt integrations. See the Recipes and Accelerators page on the Oracle Help Center.

Synchronize Comments Between GitHub Issues and a Jira Comment

You can add a comment in a GitHub issue and update the corresponding comment in Jira.

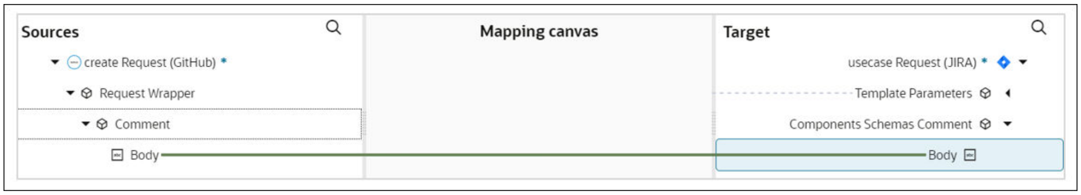
To perform this operation, you create the GitHub Adapter and Jira Adapter connections in Oracle Integration. Follow the steps below to create an integration.

1. Create an application integration.
2. Drag a GitHub Adapter into the integration canvas.
3. Configure the GitHub Adapter as follows:
   a. On the Basic Info page, provide an endpoint name.
   b. Select the business object. For this example, **Repository Events** is selected.
   c. Select the trigger operation. For this example, **Subscribe to Repository Event** is selected.
   d. On the Configuration page, select the required repository from the drop-down list and select **Issue comment Event** as the event name.
   e. On the Summary page, review your selections.
4. Drag a Jira Adapter into the integration canvas.
5. Configure the Jira Adapter as follows:
   a. On the Basic Info page, provide an endpoint name.
   b. On the Action Page, select the action type. For this example, **Create or update** is selected.

c. On the Operations page, select the **Issue comments** object and the **Add Comment** operation.

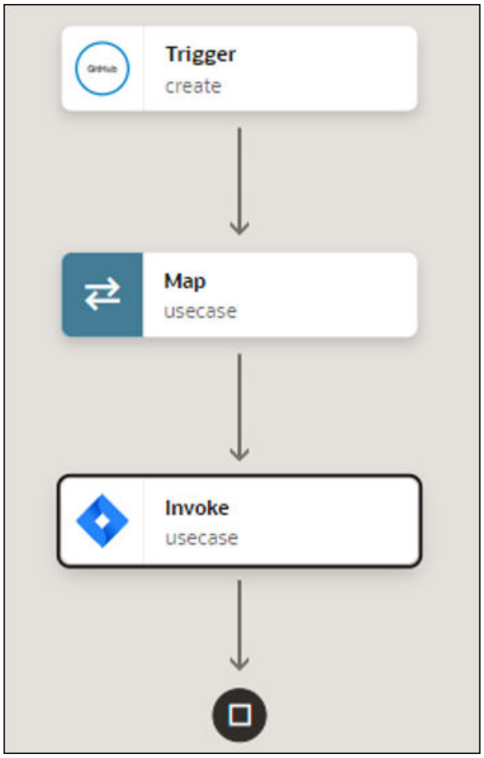
d. On the Summary page, review your selections.

6. In the mapper, map the required fields.



7. Click **Validate**.

The completed integration looks as follows.



8. When complete, save and activate the integration.

As a result, when a comment is added to an issue in GitHub, a corresponding comment is added to the related issue in Jira.