# Oracle® Cloud

# Using the Jira Adapter with Oracle Integration 3

F45560-05
April 2024

Oracle Cloud Using the Jira Adapter with Oracle Integration 3,

F45560-05

Copyright © 2022, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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.

# Contents


## Preface

Audience v
Documentation Accessibility v
Diversity and Inclusion v
Related Resources vi
Conventions vi

## 1 Understand the Jira Adapter

Jira Adapter Capabilities 1-1
Jira Adapter Restrictions 1-2
What Application Version Is Supported? 1-3
Workflow to Create and Add a Jira Adapter Connection to an Integration 1-3

## 2 Create a Jira Adapter Connection

Prerequisites for Creating a Connection 2-1
- Purchase a Jira Subscription 2-1
- Prerequisites to Use the Basic Authentication Security Policy 2-1
- Prerequisites to Use the Authorization Code Credentials Security Policy 2-2

Create a Connection 2-8
- Configure Connection Properties 2-9
- Configure Connection Security 2-9
- Configure the Endpoint Access Type 2-11
- Test the Connection 2-11

Upload a Certificate to Connect with External Services 2-12
Refresh Integration Metadata 2-14

## 3 Add the Jira Adapter Connection to an Integration

Basic Info Page 3-1
Trigger Operations Page 3-2
Invoke Action Page 3-3

Invoke Operations Page 3-3
Summary Page 3-6

4 Post-Activation Steps

Register a Webhook 4-1

5 Implement Common Patterns Using the Jira Adapter

Synchronize ServiceNow Incidents With Jira Issues 5-1
Use the Pagination Concept in an Integration 5-2
Retrieve Issue Details Using the Jira Query Language (JQL) 5-4

6 Troubleshoot the Jira Adapter

Error When Testing a JQL Query 6-1

# 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 Jira Adapter

Review the following conceptual topics to learn about the Jira 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:**


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


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

## Jira Adapter Capabilities

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

The Jira Adapter provides the following benefits:

- Supports connecting to private resources that are in your virtual cloud network (VCN) with a private endpoint. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Configure the Endpoint Access Type. This type of connection does not use the connectivity agent.
- Provides trigger (source) connection support for performing the following types of events against the selected project and issue type:
  - Comment Created
  - Comment Updated
  - Issue Created
  - Issue Updated
- Provides trigger (source) connection support for performing the following types of events:
  - Project Created
  - Project Updated
  - Project Deleted

- Provides invoke (target) connection support for performing actions such as Create or Update, Query, and Delete on the following objects:
  - Issue attachments
  - Issue comments
  - Issues
  - Projects
  - Users
- Supports Jira webhooks.
- Supports declaratively defining and executing a Jira Query Language (JQL) query.
- Supports Jira APIs protected using Basic Authentication or Authorization Code Credentials.

The Jira Adapter is one of many predefined adapters included with Oracle Integration. See the Adapters page in the Oracle Help Center.

# Jira Adapter Restrictions

Note the following Jira Adapter restrictions.

- The Jira Adapter only supports Jira Cloud. Jira Server is not supported.
- The Jira Webhooks JQL configuration (for trigger connections) has the following restrictions:

> **Note:**
>
> The following restrictions apply only to integrations created prior to the Oracle Integration 22.12 release. From the 22.12 release onwards, you can build multiple Jira trigger integrations using a single connection and activate them.
>
> - Only one webhook listener URL is allowed, per connected application per tenant. Therefore, if you want to configure multiple Jira inbound integrations, you must create multiple connections.
> - The webhook expiration period is 30 days from the time the webhook was created. See Webhook expiration. If the Jira Adapter webhook integration does not process any notifications for more than 30 days, reactivate the integration.
> - Before you migrate (import/export) a Jira trigger integration (IAR) file from one Oracle Integration instance to another instance, you must deactivate the Jira trigger integration in the Oracle Integration instance. Select the **Delete Event Subscription** checkbox in the dialog that appears when you click the deactivate icon to deactivate the integration.
> - There are other restrictions. See Supported JQL Queries.

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

# 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 Jira 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 Jira Adapter Connection |
| 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 Jira 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 in *Using Integrations in Oracle Integration 3* |
| 8 | Track payload fields in messages during runtime. | Assign Business Identifiers for Tracking Fields in Messages and Manage Business Identifiers for Tracking Fields in Messages 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 Jira 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
- Create a Connection
- Upload a Certificate to Connect with External Services
- Refresh Integration Metadata


## Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the Jira Adapter:


- Purchase a Jira Subscription
- Prerequisites to Use the Basic Authentication Security Policy
- Prerequisites to Use the Authorization Code Credentials Security Policy


### Purchase a Jira Subscription

When you subscribe to Jira, you receive a Jira instance URL, username, and password. This information is required for creating a Jira Adapter connection in Oracle Integration.

See Configure Connection Properties and Configure Connection Security.

### Prerequisites to Use the Basic Authentication Security Policy

If you want to use the Basic Authentication security policy with your Jira Adapter connection, you must perform the following prerequisites.

**Create an API Token**

1. Log in to your Atlassian account using the URL:

   ```
   https://id.atlassian.com
   ```

2. Click **Account Settings**.
3. Click **Security**.
4. Under **API token**, click **Create and manage API tokens**.
5. Click **Create API token**.
6. Enter a label and click **Create**. The API token is generated.

7. Copy the value. You'll need to enter this value in the **Password** field when you configure security for your Jira Adapter connection in Oracle Integration. See Configure Connection Security.

# Prerequisites to Use the Authorization Code Credentials Security Policy

If you want to use the Authorization Code Credentials security policy with your Jira Adapter connection, you must perform the following prerequisites.

- Create an App and Obtain the Client ID and Client Secret
- Configure the OAuth 2.0 Authorization Code and Assign the Scopes
- Minimum Scopes for Creating a Connection

### Create an App and Obtain the Client ID and Client Secret

1. Log in to your Atlassian account using the following URL:

   ```
   https://id.atlassian.com
   ```

2. Click your profile icon and select **Developer console**.
3. On the My apps page, click **Create**, and select **OAuth 2.0 integration**.
4. Enter a name for your app, select the **I agree** checkbox, and click **Create**. A new app is displayed in the **My apps** list.
5. Select your app.
6. In the left menu, click **Settings**.The client ID and client secret are displayed in the **Authentication details** section.
7. Copy the values for your client ID and client secret. You'll need to enter these values on the Connections page when you configure security for your Jira Adapter connection. See Configure Connection Security.

### Configure the OAuth 2.0 Authorization Code and Assign the Scopes

1. Open the app that you created in Jira.
2. In the left menu, click **Authorization**.
3. In the **Callback URL** field, enter the callback URL.
4. In the left menu, click **Permissions**.
5. Click **Add** next to the API you want to assign to your app.

### Minimum Scopes for Creating a Connection

These are the minimum scopes required to configure a Jira Adapter connection in an integration. Ensure that the scopes you want to provide are configured in the Jira application.

Minimum scopes to configure the Jira Adapter as a trigger connection:

- Minimum classic scopes:

  ```
  read:jira-work manage:jira-webhook
  ```

> **Note:**
>
> - Jira recommends using the classic scopes.
> - Use the granular scopes only when you can't use classic scopes.

- Minimum granular scopes:

| Event Type | Required Scopes |
|---|---|
| Issue Created | - `write:webhook:jira`<br>- `write:issue:jira`<br>- `read:webhook:jira`<br>- `read:issuemeta:jira`<br>- `read:field-configuration:jira`<br>- `read:avatar:jira`<br>- `read:issue-details:jira`<br>- `read:field.default-value:jira`<br>- `read:field.option:jira`<br>- `read:field:jira`<br>- `read:group:jira`<br>- `read:project:jira`<br>- `delete:webhook:jira`<br>- `read:jql:jira` |
| Issue Updated | - `write:webhook:jira`<br>- `read:webhook:jira`<br>- `delete:webhook:jira`<br>- `read:field:jira`<br>- `read:project:jira`<br>- `read:issue-meta:jira`<br>- `read:avatar:jira`<br>- `read:field-configuration:jira`<br>- `read:issue-details:jira`<br>- `read:field.default-value:jira`<br>- `read:field.option:jira`<br>- `read:group:jira`<br>- `write:issue:jira`<br>- `read:jql:jira` |

| Event Type | Required Scopes |
|---|---|
| Comment Created | – write:webhook:jira<br>– read:webhook:jira<br>– read:issue-meta:jira<br>– read:field-configuration:jira<br>– read:group:jira<br>– read:project:jira<br>– read:project-role:jira<br>– read:user:jira<br>– read:avatar:jira<br>– read:issue-details:jira<br>– read:field.default-value:jira<br>– read:field.option:jira<br>– read:field:jira<br>– delete:webhook:jira<br>– read:jql:jira<br>– read:comment.property:jira<br>– read:comment:jira<br>– read:epic:jira-software<br>– read:issue.property:jira<br>– read:issue-type:jira<br>– read:role:jira<br>– read:status:jira |

| Event Type | Required Scopes |
|---|---|
| Comment Updated | – read:issue-meta:jira<br>– read:avatar:jira<br>– read:field-configuration:jira<br>– write:comment:jira<br>– write:comment.property:jira<br>– write:attachment:jira<br>– read:issue:jira<br>– read:attachment:jira<br>– read:user:jira<br>– read:application-role:jira<br>– read:comment:jira<br>– read:comment.property:jira<br>– read:project-role:jira<br>– read:issue-security-level:jira<br>– read:issue.vote:jira<br>– read:issue.changelog:jira<br>– read:status:jira<br>– delete:comment:jira<br>– delete:comment.property:jira<br>– delete:attachment:jira<br>– delete:issue:jira<br>– read:issue.transition:jira<br>– write:issue.property:jira<br>– read:issue-type:jira<br>– read:project.property:jira<br>– read:issue-type-hierarchy:jira<br>– read:project-category:jira<br>– read:project-version:jira<br>– read:project.component:jira<br>– write:issue:jira<br>– read:group:jira<br>– read:project:jira<br>– read:jql:jira<br>– read:issue-details:jira<br>– read:field.default-value:jira<br>– read:field.option:jira<br>– read:field:jira |
| Project Created | – read:issue-meta:jira<br>– read:avatar:jira<br>– read:field-configuration:jira<br>– write:webhook:jira<br>– read:webhook:jira<br>– delete:webhook:jira<br>– write:project:jira<br>– read:project:jira<br>– read:field:jira |

| Event Type | Required Scopes |
|---|---|
| Project Updated | – read:issue-meta:jira<br>– read:avatar:jira<br>– read:field-configuration:jira<br>– write:webhook:jira<br>– read:webhook:jira<br>– delete:webhook:jira<br>– read:field:jira<br>– read:issue-type:jira<br>– read:project:jira<br>– read:project.property:jira<br>– read:user:jira<br>– write:project:jira<br>– write:project.avatar:jira<br>– read:application-role:jira<br>– read:group:jira<br>– read:issue-type-hierarchy:jira<br>– read:project-category:jira<br>– read:project-version:jira<br>– read:project.component:jira |
| Project Deleted | – read:issue-meta:jira<br>– read:avatar:jira<br>– read:field-configuration:jira<br>– write:webhook:jira<br>– read:webhook:jira<br>– delete:webhook:jira<br>– read:field:jira<br>– delete:project:jira |

Minimum scopes to configure the Jira Adapter as an invoke connection:

- Minimum classic scopes:
  - read:jira-user
  - read:jira-work
  - write:jira-work
  - manage:jira-project
  - manage:jira-configuration

> **Note:**
>
> – Jira recommends using the classic scopes.
>
> – Use the granular scopes only when you can't use classic scopes.

- Minimum granular scopes:
  Required scopes:
  - read:issue-meta:jira

- read:avatar:jira
- read:field-configuration:jira
- write:comment:jira
- write:comment.property:jira
- write:attachment:jira
- read:issue:jira
- read:attachment:jira
- read:user:jira
- read:application-role:jira
- read:comment:jira
- read:comment.property:jira
- read:project-role:jira
- read:issue-security-level:jira
- read:issue.vote:jira
- read:issue.changelog:jira
- read:status:jira
- delete:comment:jira
- delete:comment.property:jira
- delete:attachment:jira
- delete:issue:jira
- read:issue.transition:jira
- write:issue.property:jira
- read:issue-type:jira
- read:project.property:jira
- read:issue-type-hierarchy:jira
- read:project-category:jira
- read:project-version:jira
- read:project.component:jira
- write:issue:jira
- read:group:jira
- read:project:jira
- read:jql:jira
- read:issue-details:jira
- read:field.default-value:jira
- read:field.option:jira

- `read:field:jira`
- `read:issue-type-hierarchy:jira`
- `read:project.component:jira`
- `write:project:jira`
- `write:project.avatar:jira`
- `delete:project:jira`

# 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. |
| **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`). |

| Element | Description |
|---|---|
| **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.<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. |

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 Properties

Enter connection information so your application can process requests.

1. Go to the **Properties** section.
2. In the **Hostname** field, enter the Jira instance hostname.

## Configure Connection Security

Configure security for your Jira Adapter connection.

1. Go to the **Security** section.
2. From the **Security Policy** list, select the security policy to use:
   - **Basic Authentication**
   - **Authorization Code Credentials**
3. If you selected **Basic Authentication**:
   a. In the **Username** field, enter the username.
   b. In the **Password** field, enter the API token. An API token is used (instead of your account password) as a password whenever you programmatically authenticate with basic authentication. See Create an API Token.

> **Note:**
>
> The Basic Authentication security policy can be used for invoke connections only.

4. If you selected **Authorization Code Credentials**:
   a. In the **Client Id** field, enter the client ID that you obtained when creating an app. See Create an App and Obtain the Client ID and Client Secret.
   b. In the **Client Secret** field, enter the client secret key that you obtained when creating an app. See Create an App and Obtain the Client ID and Client Secret.
   c. In the **Scope** field, enter `read:jira-work manage:jira-webhook` for your Jira Adapter trigger connection, and enter `read:jira-work read:jira-user write:jira-work` for your Jira Adapter invoke connection. These are the minimum older scopes required to configure a connection. However, you can enter new Jira scopes. See Minimum Scopes for Creating a Connection.

      You can also enter other space-separated scopes that you need to configure. Ensure that the scopes you want to provide are configured in the Jira application. See Configure the OAuth 2.0 Authorization Code and Assign the Scopes.

> **Note:**
>
> - The Authorization Code Credentials security policy can be used for trigger and invoke connections.
> - The **Scope** field is not marked with an asterisk (*****). This incorrectly implies that this field is not mandatory. This field is mandatory.

   d. Click **Provide Consent**. You are redirected to the Jira login page.

e. Log in to the Jira account with your credentials. Once you successfully log in, the following message appears:

```
Access Allowed! Please switch back to connection page to proceed with
testing and saving
connection configuration.
```

# 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

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 Infrastructure 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 Infrastructure 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**.
- IPSec tunneling and FastConnect are not supported for use with private endpoints.

# Test the Connection

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

1. In the page title bar, click **Test**.
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, verify URLs and credentials, and download the diagnostic logs for additional details. Continue to test until the connection is successful.

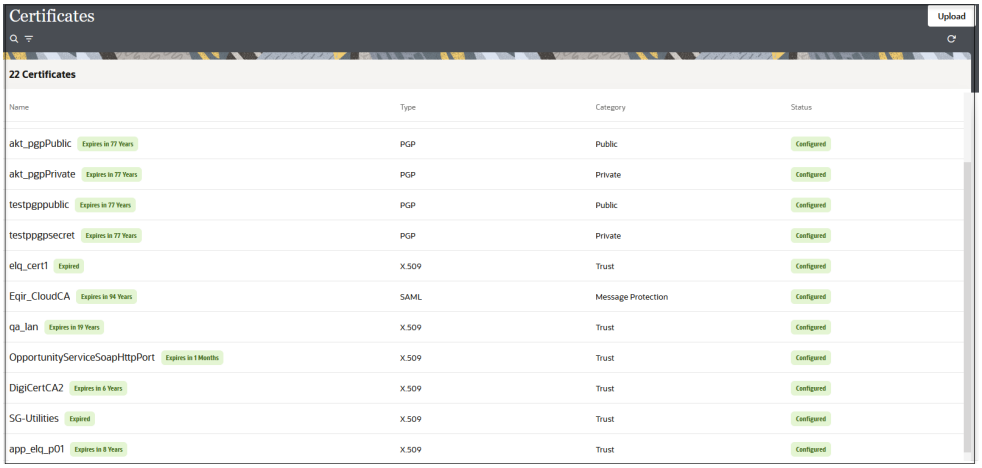
3. When complete, click **Save**.

# 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.




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 Implement Digital Signature Validation (RSA) 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 Jira Adapter Connection to an Integration

When you drag the Jira 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 Jira Adapter endpoint properties.

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

**Topics:**


- Basic Info Page
- Trigger Operations Page
- Invoke Action Page
- Invoke Operations Page
- Summary Page


## 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.` |

# Trigger Operations Page

Select the project and event you want to perform in the Jira application.

> **Note:**
>
> The **Select project**, **Select issue type**, **Webhook JQL**, and **Validate** fields are not displayed if the **Project Created**/**Project Updated**/**Project Deleted** event is selected. This is because JQL is not supported for these events.

| Element | Description |
|---|---|
| **Select project** | Select a project that you created in the Jira application. The projects you create in the Jira application are displayed for selection. |
| **Select issue type** | Select an issue type such as **Task**, **Bug**, **Story**, **Epic**, **Sub-task**, **Custom_Issue_Type**, and so on. The issue types that are displayed for selection are based on your project selection. |
| **Select Event** | Select an event such as **Issue Created**, **Issue Updated**, **Comment Created**, **Comment Updated**, **Project Created**, **Project Updated**, and **Project Deleted**. |
| **Webhook JQL** | You can enter a query or paste a query from the Jira application as per your requirement. In the query, the project and issue type field values are added based on your selection. You can select another value if required. However, don't delete these values. You can add your query next to these values.<br>Enter a valid JQL query in the text box. For example:<br>`project=Team And issueType=Task` |
| **Validate** | Click to validate the query against the Jira application. The query response is displayed for a successful query and the request fields are displayed accordingly in the request schema. If the query is invalid, you receive a response for the errors in the query. |

# Invoke Action Page

Select the type of action to perform.

| Element | Description |
|---|---|
| **Select action type** | • **Query**: Retrieves information from the Jira application corresponding to the selected object and operation.<br>• **Create or update**: Creates new records or updates the existing records in the Jira application.<br>• **Delete**: Deletes records from the Jira application. |

# Invoke Operations Page

Enter the details based on the action you selected on the Action page.

- Query
- Create or update
- Delete

> **Note:**
>
> - If you configured the Authorization Code Credentials security policy for your Jira Adapter connection, you must ensure that the respective API scope is assigned in the Jira application before you configure the related operation. Otherwise, you receive an error during runtime. See Configure the OAuth 2.0 Authorization Code and Assign the Scopes.

**Query**

| Element | Description |
|---|---|
| If you selected **Query** on the Action page to get records based on a selected object and operation or want to enter a Jira Query Language (JQL) request from/to the Jira application, the following options are displayed. | • **Retrieve**: Retrieves records based on a selected object and operation from the Jira application.<br>• **Jira Query Language (JQL)**: Enables you to define a JQL-based query to send a request for querying and retrieving the information, which is queried from the Jira application. |

The following fields are displayed if you select **Retrieve**.

| Element | Description |
|---|---|
| **Select object** | 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. |

| Element | Description |
|---|---|
| **Select operation** | Select an operation name, such as **Get Attachment**, **Search Projects**, and so on. |
| **Filter by operation name** | Type the initial letters of the operation name to filter the display of names in the list. |

The following options are displayed when you select **Jira Query Language (JQL)**.

| Element | Description |
|---|---|
| **Select project** | Select a project that you created in the Jira application. The projects you create in the Jira application are displayed for selection. |
| **Select issue type** | Select an issue type such as **Task**, **Bug**, **Story, Epic, Sub-task, Custom Issue Type**, and so on. The issue types that are displayed for selection are based on your project selection. |
| **Edit query** | Select this check box to enter a valid JQL query in the text box. For example:<br><br>`project=Demo And issueType=Sub-task OR priority=Low`<br><br>You can enter a query or paste a query from the Jira application as per your requirement. In the query, the project and issue type field values are added based on your selection. You can select another value if required. However, don't delete these values. You can add your query next to these values.<br>The query can also include custom fields and parameters. |
| **Binding parameters** | Displays any binding parameter, if included, in the specified query. For example:<br><br>`'priority'`<br><br>is a parameter in the following query:<br><br>`project=Demo And issueType=Sub-task OR priority=&priority`<br><br>Enter a query with a parameter and press the **Refresh** button above **Binding Parameters**. A text box in which to enter a test value for the parameter is displayed.<br>**Note**: If any field value contains the ampersand (`&`) symbol (for example, `project=te&st`), you must enter the ampersand (`&`) symbol (for example, `te&&st`). |

| Element | Description |
|---|---|
| **Test Query** | Click to validate the query against the Jira application. The query response is displayed for a successful query and the response fields are displayed accordingly in the response schema. If the query is invalid, you receive a response for the errors in the query. See Error When Testing a JQL Query. |

### Create or update

| Element | Description |
|---|---|
| **Select project** (displayed if you select the **Issues** object and the **Create Issue** operation) | Select a project that you created in the Jira application. The projects you create in the Jira application are displayed for selection. |
| **Select issue type** (displayed if you select the **Issues** object and the **Create Issue** operation) | Select an issue type such as **Task**, **Bug**, **Story**, **Epic, Custom Issue Type**, and so on. |
| **Select object** | 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 **Create Issue**, **Update Comment**, and so on. |
| **Filter by operation name** | Type the initial letters of the operation name to filter the display of names in the list. |

### Delete

| Element | Description |
|---|---|
| **Select object** | 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 **Remove Attachment**, **Remove User**, and so on. |
| **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 Post-Activation Steps

After you create an integration that includes a Jira Adapter as a trigger (source) connection, you must perform the following step.

> **Note:**
>
> This section only applies to integrations created after the Oracle Integration 22.12 release.

**Topics:**


- Register a Webhook


## Register a Webhook

You must register a webhook after you create an integration that includes a Jira Adapter as a trigger (source) connection.

1. Activate the integration flow in Oracle Integration and copy the endpoint URL.
2. Log in to your Atlassian account.
3. Navigate to the Jira administration console, then **System**, and then **Webhooks** (in the **Advanced** section).
4. Click **Create a webhook**.
5. Enter a webhook name.
6. In the **Status** field, select **Enabled**.
7. In the **URL** field, paste the endpoint URL that you copied in step 1.
8. In the **Issue related events** section, you must specify the JQL query that you entered on the trigger Operations page in Oracle Integration.

   > **Note:**
   >
   > Step 8 only applies to the Comment Created, Comment Updated, Issue Created, and Issue Updated events.

9. In the **Events** field, select the event that you configured in Oracle Integration.
10. Click **Create**.

5

# Implement Common Patterns Using the Jira Adapter

You can use the Jira Adapter to implement the following common patterns.

**Topics:**


- Synchronize ServiceNow Incidents With Jira Issues
- Use the Pagination Concept in an Integration
- Retrieve Issue Details Using the Jira Query Language (JQL)


> **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 ServiceNow Incidents With Jira Issues

You can synchronize ServiceNow incidents with Jira issues. The integration is invoked when the incident is created in ServiceNow and the details of the incident are added as an issue in Jira. This implementation pattern provides an overview of the steps.

1. Create an app-driven orchestrated integration.
2. Drag a ServiceNow Adapter into the integration as a trigger connection.
3. Configure the ServiceNow endpoint:
    a. On the Basic info page, provide an endpoint name, and click **Next**.
    b. On the Applications page, select the **Incident** module, and click **Next**.
    c. On the Fields page, select the required fields from **Available Fields**, and click **Next**.
    d. On the Conditions page, select **Created**, and click **Next**.
    e. On the Summary page, review your selections, and click **Done**.
4. Drag a Jira Adapter into the integration canvas.
5. Configure the Jira 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 the **Issues** object, project, issue type, and then select the **Create Issue** operation. Click **Next**.

d. Review your selections on the Summary page, and click **Done**.

6. In the mapper, perform the required mappings to create an issue in Jira.
7. Drag a switch action and specify the condition if the issue is created in Jira.

```
If id != ""
```

8. Drag a second ServiceNow Adapter into the integration canvas and configure it with the **Create, Update Or Delete Information** action.
   a. On the Basic info page, provide an endpoint name, and click **Next**.
   b. On the Action page, select **Create, Update or Delete Information**, and click **Next**.
   c. On the Operations page, select the **Update an existing record in the selected Module or Table, identified by the mandatory sys_id field** operation and select the **Incident** module. Click **Next**.
   d. On the Summary page, review your selections, and click **Done**.
9. Update the **Additional comments** field as a response in ServiceNow (that is, **Successful** when the issue gets created in Jira).
10. Activate the integration.
    a. At the end of the row for the integration, click the **Switch to Activate** icon.
    b. If needed, check the **Enable Tracing** check box. The **Include Payload** check box is displayed.
    c. If needed, check the **Include Payload** check box.
    d. Click **Submit Now**. The integration is deployed.

# Use the Pagination Concept in an Integration

When you must fetch a large number of results in Jira, you can use the Jira Adapter's pagination feature. Pagination helps you segregate the total number of results. You can specify the number of results per page according to your requirement while triggering an integration and receive sorted results.

This use case describes how to use pagination to sort results while fetching product results. Similarly, you can use pagination for the Issue attachments object, Issue comments object, and Users object. To perform this operation, you create an integration for the Jira Adapter in Oracle Integration using pagination with the orchestrator.

1. Create a scheduled orchestrated integration.
2. Drag an assign action to the workspace.
3. Provide a unique name for this action, and click **Create**.

4. Assign the following two variables to your integration. Specify the values to variables as follows.

```
$startAt="0"


$Max ="2"
$Flag="false"
```

5. Click **Validate**, and then click **Close**.
6. Drag a while action below the assign action.
7. Provide a unique name for this action, and click **Create**.
8. Set the condition for the specified variables as follows.

```
$Flag !="true"
```

9. Click **Validate**, then click **Close**.
10. Drag a Jira Adapter inside the while action.
11. Specify the following details in the Adapter Endpoint Configuration Wizard.
    a. On the Basic Info page, provide an endpoint name, and click **Next**.
    b. On the Actions page, select **Query**, and click **Next**.
    c. On the Operations page, select the **Projects** as an object, and select **Search Projects** as an operation.
    d. On the Summary page, review your selections, and click **Done**.
12. In the mapper, map **$Max** to **maxResults** and **$startAt** to **startAt**.
13. Drag an FTP Adapter connection and configure the FTP endpoint to write the response to a file.
    a. On the Basic Info page, provide a name, and click **Next**.
    b. On the Operations page, select **Write File** from the **Select Operation** list.
    c. Select **Binary** in the **Select a Transfer Mode** field.
    d. Specify the output directory, file name pattern, and download directory, and click **Next**.
    e. On the Scheme page, select **No** in the **Do you want to specify the structure for the contents of the file** field, and click **Next**.
    f. Review your selections on the Summary page, and click **Done**.
14. Perform the required mapping.
15. Drag a switch action and specify the following IF condition:

```
contains( nextPage, "startAt") = "true"
```

16. Drag an assign action to update the specified variables as follows:

```
startAt + $Max
```

17. Drag an assign action into the **Otherwise** branch of the switch action to update the specified variables as follows:

    ```
    $Flag="true"
    ```

18. When complete, save and activate the integration.
19. After successful activation, you can submit the integration and monitor the runtime in Oracle Integration. The integration keeps running until it completes all cycles of fetching project results, respectively, to each page in the FTP location.

# Retrieve Issue Details Using the Jira Query Language (JQL)

The Jira Adapter enables you to define a Jira Query Language (JQL)-based query to send a request to query and retrieve issue details from the Jira application. This use case describes how to use the Jira Adapter to send a JQL query and write the response to a target file location.

In this use case, an FTP Adapter is used to write the data that you receive from Jira to a file. However, you can also use other applications.

1. Create a scheduled orchestrated integration.
2. Drag and drop an assign action into the integration canvas and create three variables:
    - `$startAt=”0”`
    - `$Max=”2”`
    - `$Flag=”false”`
3. Drag a while action and specify the condition as follows:

    ```
    $Flag!=”true”
    ```

4. Drag a Jira Adapter inside the while action and configure it with the JQL operation.
    a. On the Basic info page, provide an endpoint name, and click **Next**.
    b. On the Action page, select **Query**, and click **Next**.
    c. On the Operations page, select **Jira Query Language (JQL)**.
    d. Select a project, an issue type, and enter or paste a query.
    e. If required, add a binding parameter in the query, click **Refresh**, and enter a value for the parameter in the text box.
    f. Click **Test Query** to validate the query, then click **Next**.
    g. On the Summary page, review your selections, and click **Done**.
5. In the mapper, map the **$maxResults** variable to **maxResults** and the **$startAt** variable to **startAt**.
6. Drag an FTP Adapter and configure it to write the data that you receive from Jira.
    a. On the Basic info page, provide an endpoint name.
    b. On the Operations page, select **Write File** from the **Select Operation** list.
    c. Select **Binary** from the **Select a Transfer Mode** list.
    d. Provide the output directory and file name pattern.

e. On the Schema page, select **No**.

f. On the Summary page, review your selections, and click **Done**.

7. Perform the required mapping.
8. Drag a switch action and specify the following IF condition.

```
issues != ""
```

9. Drag an assign action to update the specified `$startAt` variable as follows:

```
startAt + $Max
```

10. Drag an assign action into the **Otherwise** branch of the switch action to update the specified variable as follows:

```
$Flag="true"
```

11. When complete, save and activate the integration.
12. After successful activation, you can submit the integration and monitor the runtime behavior in Oracle Integration. The integration keeps running until it completes all cycles of fetching results, respectively, to each page in the FTP location.

    Further in your orchestration, you can add adapters and actions as per your business requirement.

6

# Troubleshoot the Jira Adapter

Review the following topics to learn about troubleshooting issues with the Jira Adapter.

**Topics:**


- Error When Testing a JQL Query


Additional integration troubleshooting information is provided. See Troubleshoot Oracle Integration in *Using Integrations in Oracle Integration 3* and the Oracle Integration Troubleshooting page on the Oracle Help Center.

## Error When Testing a JQL Query

The following error occurs when you test a Jira Query Language (JQL) query on the Operations page of the Adapter Endpoint Configuration Wizard and there is any null value present in the JQL result.

```
(issuelinks) Array element is empty. Please add more than one element in
this
array and then do the json sampling. Check the error stack and fix the cause
of the error.
Contact oracle support if error is not fixable.
```

**Solution**: Replace all the null values (that is, `[]` ) with double quotation marks (`""`) in the JQL result.