Oracle® Cloud

Using Agentic AI in Oracle Integration 3

G41284-01

Oracle Cloud Using Agentic AI in Oracle Integration 3,

G41284-01

Primary Author: Oracle Corporation

Contents

About This Content

Using Agentic AI in Oracle Integration 3 describes how to automate your workflows using AI agents and agentic AI tools in Oracle Integration.

Audience

Using Agentic AI in Oracle Integration 3 is intended for users who want to automate workflows with AI agents and agentic AI tools in Oracle Integration.

Documentation Accessibility

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

Access to Oracle Support

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

Related Resources

For more information, see these Oracle resources:

- 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

Get Started

Get started with Oracle Integration and agentic AI.

Topics:

How Integrations and AI Agents Fit Together

You use AI agents in Oracle Integration when you need to orchestrate different integrations in an adaptable and flexible automation. You're looking to achieve a complex business goal. AI agents orchestrate integrations as tools to fit your business needs.

Integrations Define a Set Order of Steps

With integrations, you define a specific order of steps. The steps are run the exact same way every single time. You use integrations when you want to control the exact steps and order, and to get enterprise connectivity.

In some cases though, you may have too many possibilities and designing an integration would be too complex. For those cases, you can use AI agents.

Agents Adapt: the Integrations Used Depend on the Goal

An AI agent is a software program that uses a Large Language Model (LLM) to reason without human intervention to achieve a specific goal. AI agents decide which steps to take and in which order, depending on what is happening in the environment. AI agents are more adaptable.

Integrations are Tools for AI Agents

AI agents use tools to communicate with the world. In Oracle Integration, integrations become agentic AI tools. AI agents use integrations as agentic AI tools to connect to the external world. AI agents determine which integrations to use and in which order to achieve the goal that you define. You create AI agents in projects.

AI Agents Use Thinking Patterns to Reason and Make Decisions

AI agents use specific thinking patterns to reason and make decisions. Patterns that are available by default are:

- ReAct
- Plan and Execute

You can also customize the existing thinking patterns or create your own for your AI agents.

Integrations are Discoverable from MCP Clients and Agent Frameworks that Support MCP

You can create AI agents in Oracle Integration, or you can create AI agents in other agent frameworks such as AI Agent Studio for Fusion Applications, Langflow, or others. Discover and use integrations as tools in any third-party agent framework that supports MCP.

Regardless of the agent framework that you use, any integration can be used as an agentic AI tool. You can define which integrations in a projects to expose as tools as an MCP server. Each project becomes an MCP server.

Involve Humans for Approval with Human in the Loop

Human in the loop enables human intervention in key decision areas, when errors occur, and to ensure quality and reliability. Human in the loop ensures people are always in control.

For example, you have an AI agent that automates employee onboarding. If important documents are missing or a background check raises a flag, the AI agent can ask Human Resources for review before continuing.

You use Human in the loop:

- When approval is needed for an AI agent to execute a particular tool
- When the AI agent needs approval before doing something else
- When the AI agent doesn't know what to do. Instead of ending, the AI agent can ask a human what the next steps should be
- For error handling, when an AI agent calls a tool that results in an error, the agent can tell a human about it who can provide feedback on what to do

Corporate Documents become Knowledge Bases for Agent Actions

Corporate documents are no longer silos and difficult to find and consult. Any corporate document can become a knowledge base that AI agents and humans can query and use as reference. You add documents to a RAG knowledge base that exists within Oracle Integration and query those documents from within an integration, or use the knowledge base as a tool for AI agents.

2

Using Integrations as Tools in an MCP Server

You can invoke integrations from any Model Context Protocol(MCP) client by registering the integration as a tool, then enabling MCP for the project.

Topics:

FAQs for Projects as MCP Servers

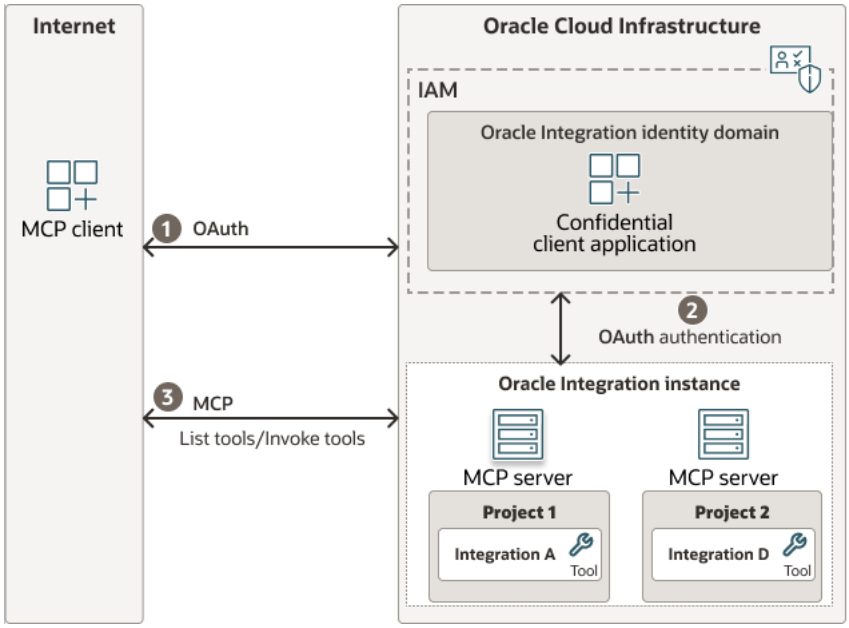
Projects can be configured to act as Model Context Protocol(MCP) servers for agentic AI tools.

How do projects become MCP servers?

You can invoke integrations from any MCP client by registering the integration as an agentic AI tool, then enabling MCP for the project. Each project becomes an MCP server and has its own MCP server URL.

Only integrations registered as agentic AI tools are discoverable through MCP.

Security is provided through OAuth.



How many MCP severs are there?

There is one MCP server per Oracle Integration project.

Supported operations are `tools/list` and `tools/call`.

Each project has its own MCP server URL. The `IntegrationInstance` parameter is required.

For example:

```
https://design.integration.us-phoenix-1.ocp.oraclecloud.com/ic/api/
integration/v1/mcpservers/projects/PROJECT_A/mcp?
integrationInstance=myInstance
https://design.integration.us-phoenix-1.ocp.oraclecloud.com/ic/api/
integration/v1/mcpservers/projects/PROJECT_B/mcp?
integrationInstance=myIinstance
```

What kind of security is there?

You must use OAuth 2.0 to expose integrations as tools with MCP. You need to create a confidential client application, assign scopes and roles, and activate it. See Complete Prerequisites: Create and Activate the Client Application.

- You must be the OCI tenant and domain administrator to configure the confidential client application.
- You need at least one confidential client application per Oracle Integration instance.
- The confidential client application must be assigned the ServiceInvoker role. For a detailed description of user roles, see What Users Can Do in the Integrations Design Section by Role.

Can any integration become a tool?

Any integration can be used as an Agentic AI tool, but the integration must meet the following criteria:

- The integration must be part of a project.
- The integration must be Active.
- The first connection in your integration must be a REST trigger connection with the REST Adapter.
- The REST trigger connection in your integration must have:
  - Authentication type OAuth.
  - JSON payload
  - POST verb

Is MCP available in all regions?

Yes, MCP is available in all Oracle Integration regions. See Availability.

Where do I find the MCP server URL?

You can find the MCP server URL in the project details page. If the project has MCP server enabled, you'll see the MCP server URL. For instructions, see Get the MCP Server URL.

Workflow to Use an Integration as a Tool with MCP

Here's a summary of the steps to follow to use an integration as an agentic AI tool and discover it through Model Context Protocol (MCP).

Step	Task
Prerequisites	Complete prerequisites: 1. Find out more about how projects can be MCP servers. See FAQs for Projects as MCP Servers . 2. Create a confidential application. Before you can discover an integration as a tool through MCP, you need to configure a confidential application to connect with OAuth. For instructions, see Complete Prerequisites: Create and Activate the Client Application. Step summary: a. Create a confidential application. b. Assign scopes to the confidential application. c. Assign the ServiceInvoker role to the confidential application. d. Activate the confidential application. e. Get the confidential application client ID and secret. f. Get the confidential application access token.
1	Register an Integration as an Agentic AI Tool
2	Enable MCP for the Project
3	Get the MCP Server URL
4	Discover Integrations as Tools from MCP Clients

Complete Prerequisites: Create and Activate the Client Application

Configure and activate the confidential client application, then take note of the client ID and secret. You'll need that information to connect from third-party applications.

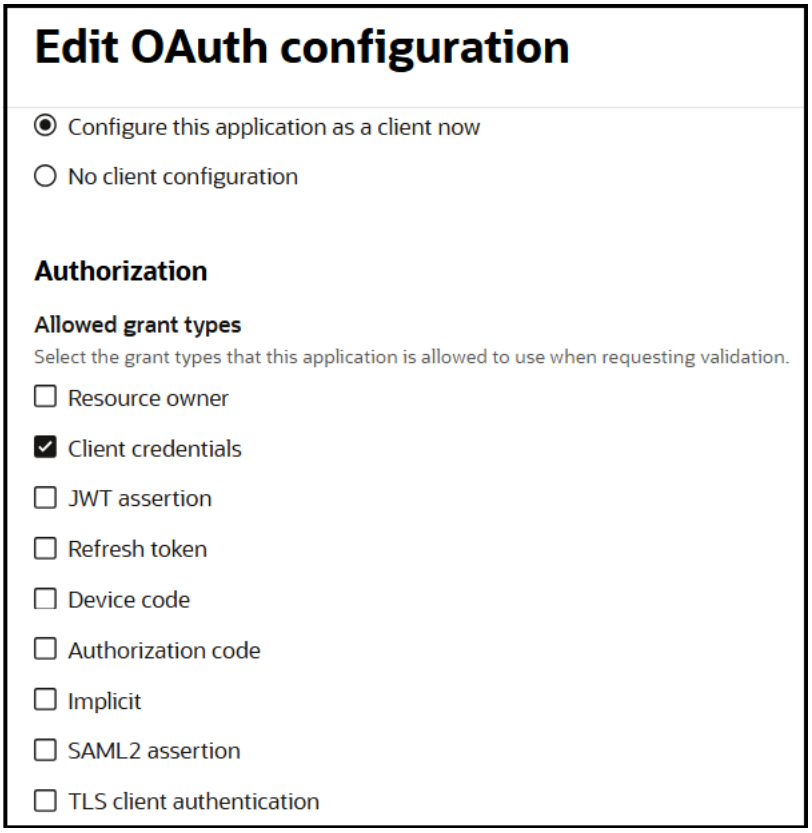
When you configure the confidential client application, you specify the grant type and assign scopes and roles. The confidential client application requires the ServiceInvoker role.

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

      This is the location at which you create the client application for your grant type.

2. Create and configure the client application.
   a. Click **Add application**.
   b. Select **Confidential Application**, then click **Launch workflow**.
   c. Enter a **name**.

      The remaining fields on this page are optional and can be ignored.
   d. Click **Submit**.
   e. Click the **OAuth configuration** tab, then the **Edit OAuth configuration** subtab.
   f. In the **Client configuration** panel, select **Configure this application as a client now**.
   g. For client credentials, select **Client credentials** in the **Allowed grant types** section



h. Leave the **Redirect URL**, **Post-logout redirect URL**, and **Logout URL** fields blank.

i. For **Client type**, ensure that **Confidential** is selected.

j. Bypass several fields and scroll down to the **Token issuance policy** section.

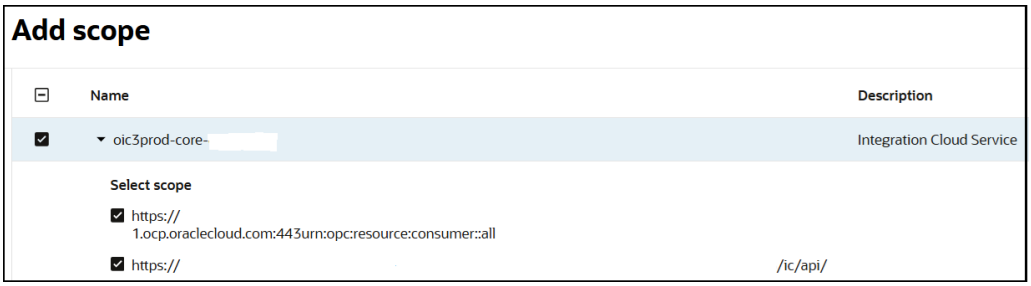
k. Select **Confidential** in the **Authorized resources** section.

l. Click the **Add Resources** toggle.

m. Click **Add scope**.

n. Find and expand the Oracle Integration application for your instance.

o. Select the two scopes appended with the following details: **urn:opc:resource:consumer::all** and **ic/api/**.



p. Click **Add**.

The scopes are displayed in the **Resources** section.

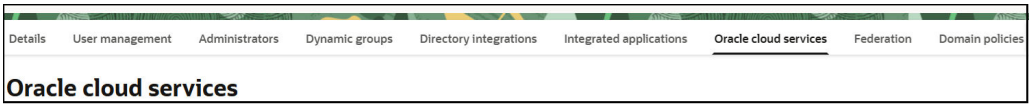
q. Ignore the **Add app roles** check box. This selection is not required.

r. Click **Submit**.

The details page for the client application is displayed.

3. Add the **ServiceInvoker** role to the client application.

a. In the menu bar, click **Oracle cloud services**.

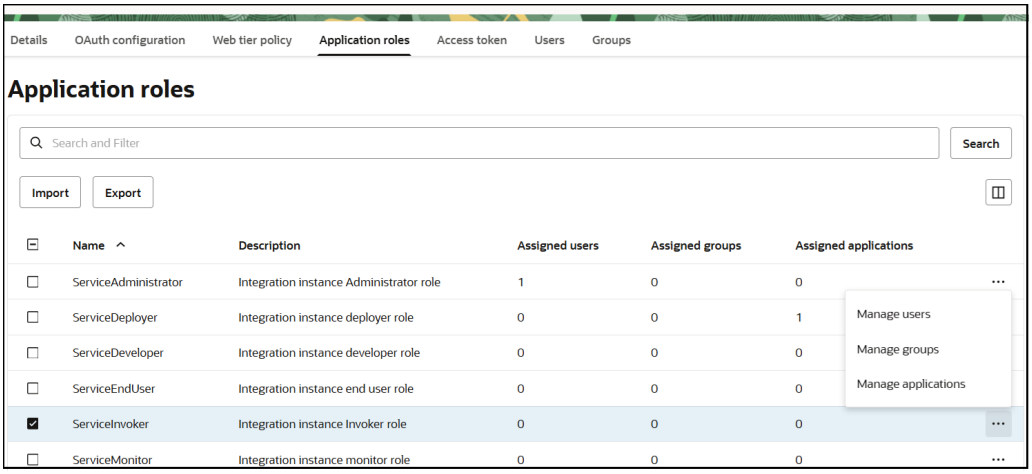


b. Click the specific application corresponding to the Oracle Integration instance.

c. In the menu bar, click **Application roles**.

d. For client credentials, expand **ServiceInvoker**, then click **Actions** ⋮ next to **Assigned applications**.

Select to assign users, groups, and applications to the instance application.

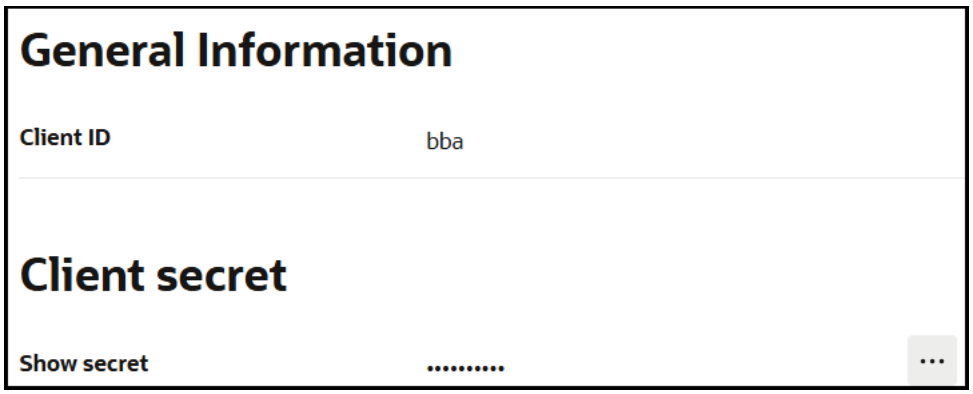


4. Activate the confidential application.

- From the **Actions** menu at the top, select **Activate**, and then **Activate application** to activate the client application for use.

5. Get the confidential client application client ID and secret.

In the **General Information** section, note the client ID and client secret values. These values are required for the third-party application that is communicating with the identity domain.

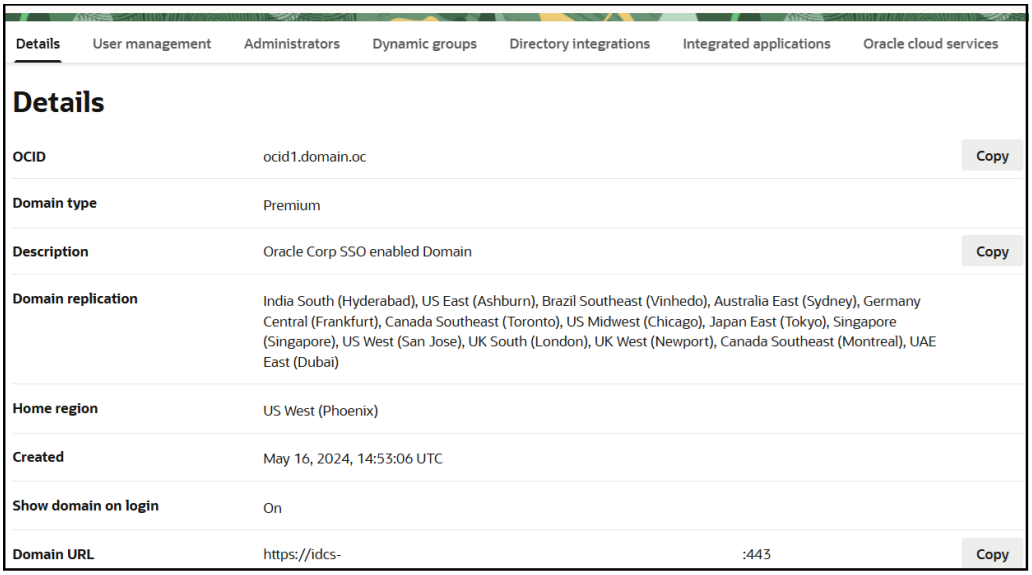


6. Get the access token for the client credentials grant type.
   a. Fetch the access client to make an access token request with the client credentials.

```
##Syntax
curl -i -H 'Authorization: Basic <base64Encoded clientid:secret>' -H
'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --
request POST https://
<Identity_Domain_Service_Instance>.identity.oraclecloud.com/oauth2/v1/
token -d 'grant_type=client_credentials&scope=<app scope>'
###where
#### <base64-clientid-secret> - Base 64 encode clientId:ClientSecret
#### <app scope> - Scope added while creating application in client
configuration section (Ends with urn:opc:resource:consumer::all)

##Example
curl -i -H 'Authorization: Basic OGQyM...ZDA0Mjcz' -H 'Content-Type:
application/x-www-form-urlencoded;charset=UTF-8' --request POST https://
<identity_domain_host>/oauth2/v1/token -d
'grant_type=client_credentials&scope=https://<Resource APP
Audience>urn:opc:resource:consumer::all'
```

   Where `Identity_Domain_Service_Instance` is the value in the **Domain URL** field of the **Details** tab of the instance application.



b. Capture the `access_token` from the response to use for authorization.

```
{
    "access_token": "eyJ4NXQjG...dfsdfsFgets2ed",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Next Step: Register an Integration as an Agentic AI Tool

Register an Integration as an Agentic AI Tool

An integration can become an agentic AI tool for AI agents. AI agents can then invoke the integration as a tool to accomplish a specific task.

Prerequisites:

Any integration can be used as an Agentic AI tool, but the integration must meet the following criteria:

- The integration must be part of a project.
- The integration must be Active.
- The first connection in your integration must be a REST trigger connection with the REST Adapter.
- The REST trigger connection in your integration must have:
  - Authentication type OAuth.
  - JSON payload
  - POST verb

1. In the navigation pane, select **Projects**.
2. Select the project in which your integration is located.
3. In the **Integrations** section, find the integration that you want register as a tool.

4. Check that your integration is Active. If it's not active, activate it by clicking **Actions** ⋮, and selecting **Activate**.
5. Register the integration as a tool. Click **Actions** ⋮, and select **Create agentic AI tool**

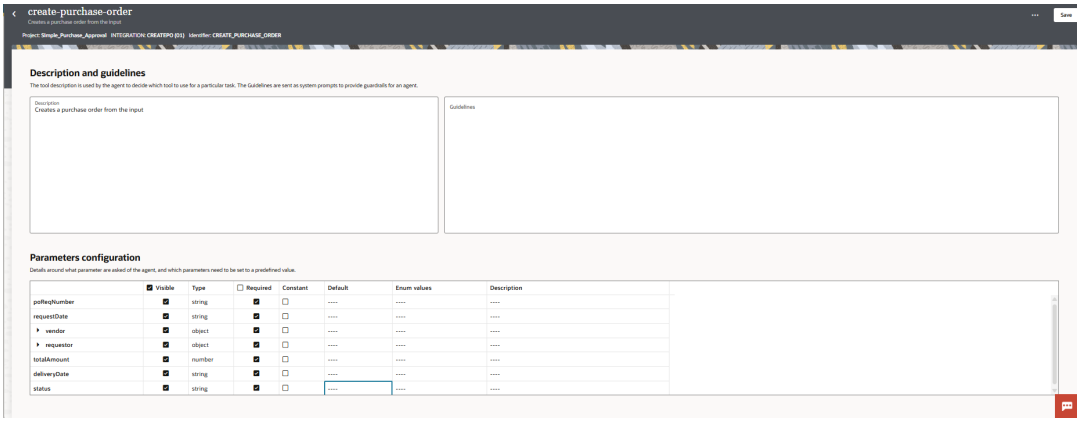
   The Create Tool panel is displayed.
6. Enter information for the tool.

Field	Description
Name	Required. Not sent to the Large Language Model (LLM). Automatically populated from the integration name. Example: `validate-invoice-data`. You cannot change the name after the tool has been created.
Identifier	Required. Uniquely identifies the tool in the project. By default, the identifier is automatically populated from the tool name. Sent to the LLM as tool metadata. The AI agent uses this information to understand the purpose of the tool and when to use it. Specify a descriptive tool identifier so that the AI agent knows exactly what the tool does. Example: `validate-invoice-data`. You cannot change the identifier after the tool has been created
Description	Required. By default, automatically populated from the integration description. Sent to the LLM as part of the system prompt. The description helps the LLM decide when to use the tool. Add information to clearly describe what the tool does and when to use it. Clear descriptions help LLMs use tools correctly and reduce errors. Example: `Validates invoice data against business rules and vendor database. Usage: Use when processing invoices to ensure data accuracy before approval.`

7. Click **Create**.

   The Tool details page is displayed. The tool lists the tool description, guidelines, and input parameters.
8. Enter additional information for the tool and identify which parameters are sent to the LLM.

   You already defined the description.



Field	Description
Guidelines	Optional. Sent to the LLM as part of the system prompt. Guidelines are constraints to limit tool behavior and respect corporate policies. Guidelines influence the LLM decision process. Specify constraints such as when the tool should be used and when it should not be used. For example, if you had a weather tool and you wanted to limit queries to only cities in the United States, you could specify as a guideline: `only use the tool for cities in U.S.`
Parameters configuration	Lists all input parameters for the integration. These are automatically populated. Configure parameter information for the tool. • **Visible:** Add a checkmark to the fields that will be sent to the LLM. Fields that do not have a checkmark are not sent to the LLM. – Expose only essential parameters. – Do not send to the LLM technical or internal parameters that are not relevant to the purpose of the tool. • **Type:** Automatically populated from the integration. • **Required:** When you indicate that a field is required, you are indicating to the LLM that it must assign a value to that parameter. Specifying a parameter as required ensures that there will always be a value for the parameter when the AI agent calls the agentic AI tool. • **Constant:** Enter any specific value that you want to assign to the parameter and you do not want the LLM to decide on. • **Default Values:** Specify default values for parameters when you know there's a safe value for the parameter if no value is assigned. This reduces the chance of LLM hallucination. • **Enum Values:** Specify comma-separated values when there are several options that could be sent to the LLM. For example, for temperature, you would specify: `celsius,fahrenheit`. • **Description:** Required. Describe all parameters clearly. The LLM depends on the descriptions to understand the tool. Clear descriptions reduce LLM errors.

9. Click **Save** to save your changes.

Next step: Enable MCP for the Project

Enable MCP for the Project

When you enable MCP for a project, the project becomes an MCP server. Any integrations registered as agentic AI tools are discoverable through the MCP server URL, and AI agent frameworks that support MCP can invoke the integrations. Each project has its own MCP server URL.

For additional details on how MCP works with projects, see FAQs for Projects as MCP Servers .

Prerequisites: Register an Integration as an Agentic AI Tool

1. In the navigation pane, select **Projects**.
2. Select the project for which you want to enable MCP.
3. In the upper right corner, click  to display the Project details.
4. Click **Enable MCP server**.
5. Click **Save changes**.

   The MCP server URL is created when you save the project.
6. Click  to display the Project details again.

   The MCP server URL is listed. You can now use the MCP server URL to invoke integrations from AI agent frameworks and MCP clients.

Next Step: Discover Integrations as Tools from MCP Clients

Get the MCP Server URL

You can find the MCP server URL in the project details page. If the project has MCP server enabled, you will see the MCP server URL.

For additional details on how MCP works with projects, see FAQs for Projects as MCP Servers .

1. In the navigation pane, click **Projects**.
2. Select your project.
3. In the upper right corner, click  to display the Project details.

   The MCP server URL is listed.

Next Step: Discover Integrations as Tools from MCP Clients

Discover Integrations as Tools from MCP Clients

To discover integrations as agentic AI tools from MCP clients or AI agent frameworks that support MCP, you need to specify the MCP server URL and use the transport mechanism streamable HTTP.

Prerequisites:

You need to create a confidential client application, register an integration as an agentic AI tool, enable MCP for the project, and get the MCP server URL.

1. Complete Prerequisites: Create and Activate the Client Application
2. Register an Integration as an Agentic AI Tool
3. Enable MCP for the Project
4. Get the MCP Server URL

Information You Need to Connect to a Project's MCP server

MCP server URL:

- How to get it: see Get the MCP Server URL.
- Format:

```
https://<OIC_HOST_NAME>/ic/api/integration/v1/mcpServers/projects/
<PROJECT_IDENTIFIER>/mcp?integrationInstance=<OIC_INSTANCE_NAME>
```

  Example:

```
https://design.integration.us-phoenix-1.ocp.oraclecloud.com/ic/api/
integration/v1/mcpservers/projects/PROJECT_A/mcp?
integrationInstance=myInstance
```

Access token for the confidential client application. See Complete Prerequisites: Create and Activate the Client Application to get the information.

Transport mechanism:

- streamable HTTP

Connect to the Project MCP Server with an MCP Client

Here's an example of connecting to a project's MCP server with Postman. You configure OAuth Authorization in Postman and add the MCP Server URL.