

Get Started with the Rapid Adapter Builder in Oracle Integration 3

In this quick-start guide, you'll learn the basics of the Rapid Adapter Builder in Oracle Integration and build a sample adapter for an application.

If you're new to Oracle Integration, familiarize yourself with its key concepts (such as, integrations, connections, mappings, and so on) before beginning this tutorial. See Integration Concepts and About Connections in *Using Integrations in Oracle Integration 3*.

Note:

- This tutorial requires access to Oracle Cloud. To sign up for a free account, see [Get started with Oracle Cloud Infrastructure Free Tier](#).

About the Rapid Adapter Builder

The Rapid Adapter Builder in Oracle Integration enables you to build an adapter for any application that exposes RESTful APIs, without having to develop complex code from scratch.

In Oracle Integration, an adapter provides an interface to connect an integration flow with a specific application. Adapters simplify integrations by eliminating the complexity associated with connectivity methods. You can use an adapter to create connections, configure invoke and trigger endpoints, and support runtime activities while designing an integration.

Oracle Integration has an ever-growing library of application-specific adapters that you can readily use for your integration scenarios. However, when an Oracle-provided adapter is not available for your purpose, you can build your own adapter using the Rapid Adapter Builder. It provides all the necessary infrastructure to build adapters for Oracle Integration. An adapter built using the Rapid Adapter Builder can offer the same capabilities as an Oracle-provided adapter. You can implement behaviors similar to those available in the existing adapters on Oracle Integration.

The Rapid Adapter Builder is available as a Visual Studio Code (VS Code) extension, which helps you generate the code required to build an adapter. Using this extension, you can iteratively develop your adapter and publish it to Oracle Integration.

About the Adapter Definition Document

To publish an adapter to Oracle Integration, you must define the specifications and functions of your adapter in a JSON-metadata file, termed as the adapter definition document.

An adapter definition document typically describes the design and implementation of an adapter. It contains information about the adapter's behavior and properties. Using the Rapid Adapter Builder, you can quickly create an adapter definition document from a Postman collection of an application's APIs (or an OpenAPI document describing a set of application APIs), and then you can modify the adapter definition document manually according to your requirements.

An adapter definition document consists of the following sections:

Section	Description
info	Describes the general information and branding of the adapter.
schemas	Defines the structure that appears in the mapper in Oracle Integration.
connection	Defines the connection parameters and security policies supported by adapter.
triggers	Defines the events that the adapter can receive to start an integration.
actions	Defines the operations that can be implemented against the target application. Actions provide a user-centric encapsulation of the operations available and exposed in the target application.
categories	Defines grouping of actions and triggers for a better user experience.
flows	Defines the implementation logic that drives the adapter. Flows are modeled using a subset of the constructs that are supported in the Serverless Workflow specification of the Cloud Native Computing Foundation (CNCF) project.

Build a Sample Adapter

In this section, you'll learn how to quickly build a sample adapter using the Rapid Adapter Builder.

For the purpose of this tutorial, let's build an adapter for the [OpenWeather](#) application. The adapter uses the public APIs provided by OpenWeather to retrieve the weather information for a given city. After you build this adapter, you'll register it on Oracle Integration.

 **Note:**

This tutorial illustrates building an adapter using a Postman collection of an application's APIs. However, you can also build an adapter using an OpenAPI document (describing a set of application APIs). For more information, see *Workflow for Building and Testing an Adapter in Using the Rapid Adapter Builder with Oracle Integration 3*.

Perform the following tasks to build your first adapter:

1. [Complete the Prerequisites](#)
2. [Construct the API Request for OpenWeather](#)
3. [Create a Postman Collection Using the API Request](#)
4. [Generate the Adapter Definition Document](#)
5. [Validate the Adapter Definition Document](#)
6. [Register the Adapter on Oracle Integration](#)

Complete the Prerequisites

Before you can start building the adapter, you must complete a few prerequisite tasks, such as, obtaining access to Oracle Integration, installing the required software, and so on.

Order	Task	More information
1	Get access to Oracle Integration	If you don't have them already, get: <ul style="list-style-type: none">• An account on Oracle Integration, including your sign-in credentials• The URL for your Oracle Integration instance Work with your Oracle Integration administrator to complete this task.
2	Create a client application in IDCS for your Oracle Integration instance and obtain the client credentials	See Create a Client Application and Obtain Credentials .
3	Assign the integration-instance developer role to the client application you created previously	See Assign the Integration-Instance Developer Role to the Client Application .
4	Download and install the required software	See Download and Install the Required Software .
5	Install the VS Code extension for Rapid Adapter Builder	See Install the VS Code Extension for Rapid Adapter Builder .

Order	Task	More information
6	Configure the VS Code extension for Rapid Adapter Builder	See Configure the VS Code Extension for Rapid Adapter Builder .

Create a Client Application and Obtain Credentials

In Oracle Identity Cloud Service (IDCS), create a client application for your Oracle Integration instance and obtain the client credentials.

You'll use these client credentials in your publisher-profile details while publishing an adapter using the Rapid Adapter Builder.

To create an IDCS application and obtain the required details, see *Prerequisites for Client Credentials* in *Using OAuth 2.0 Grants in Oracle Identity Cloud Service Environments*.

You must obtain the following details for your Oracle Integration instance:

- `Host`: URL of the Oracle Integration instance
- `integrationInstance`: Name of the Oracle Integration instance
- The following client credentials:
 - `tokenUrl`
 - `clientId`
 - `clientSecret`
 - `scope`

Assign the Integration-Instance Developer Role to the Client Application

To successfully use the client credentials, assign the integration-instance developer role to the client application you created previously.

To assign the required role to the client application:

1. Log in to the Oracle Identity Cloud Service (IDCS) console.
2. Click **Oracle Cloud Services**.
The Oracle Cloud Services page appears.
3. Search for your Oracle Integration instance and select it.
The details of the instance appear.
4. Click **Application Roles**.
The list of all applicable roles appears.
5. Locate and click the **ServiceDeveloper** role, and then click **Assign Applications**.

The Assign Applications dialog appears, displaying the client application you created previously.

6. Select the client application and click **OK**.

Now, IDCS assigns the integration-instance developer role to the application.

Download and Install the Required Software

To use the Rapid Adapter Builder, you require additional applications or software installed on your system.

1. Download and install the Postman application. Visit the [Download Postman](#) page. Create an account on Postman to access full features.
2. Download and install VS Code. Visit the [Download Visual Studio Code](#) page. You must use VS Code version 1.73.0 or higher.

Install the VS Code Extension for Rapid Adapter Builder

Download and install the VS Code extension for Rapid Adapter Builder.

1. On Visual Studio Marketplace, visit Oracle's [page](#).
2. Find and click the **Oracle Integration Cloud Rapid Adapter Builder** tile. The Details page opens.
3. Click **Install**. The extension file opens in VS Code.
4. Click **Install** in VS Code.

Now, the extension is installed and appears on the sidebar.

Configure the VS Code Extension for Rapid Adapter Builder

After you install the VS Code extension for Rapid Adapter Builder, you must initialize a workspace on your system and configure the publisher profile for the extension.


Select a folder on your system to use as a structured workspace and organize adapter-related artifacts within it. Initialize this workspace using the VS Code extension.

Additionally, update the publisher profile of the extension with the details of the Oracle Integration instance that you'll publish the adapter to. You'll enter the client credentials you obtained earlier along with some additional data. The information in the publisher profile is used to communicate with the specified Oracle Integration instance for various tasks. The extension securely stores the client credentials you enter for each Oracle Integration instance in the `publisher-profiles.yaml` file. The structure of the `yaml` file is as follows:

Property Name	Definition
active	The active profile to use. The value must match a profile name from the profiles list.
profiles	The array of profile objects.
profiles[*].name	The unique name for a profile.
profiles[*].host	The design-time URL of the Oracle Integration instance for REST API.
profiles[*].integrationInstance	The name of your Oracle Integration instance.
profiles[*].auth.tokenUrl	The IDCS get-token URL for your Oracle Integration instance.
profiles[*].auth.clientId	The client ID of the configured application in IDCS.
profiles[*].auth.clientSecret	The client secret of the configured application in IDCS.
profiles[*].auth.scope	The allowed scope of the configured application in IDCS.

Perform the following tasks in VS Code to make the required configuration.

1. Initialize a workspace using the VS Code extension.

- a. In VS Code, click **Explorer**  on the sidebar.
- b. Click **Open Folder**, and choose a folder on your system as the workspace for adapter development.

The folder is now listed on the left in the Explorer view.

 **Note:**

You can also click **File** in the top ribbon, and select **Open Folder...** to open a folder in VS Code.

- c. Press **Ctrl + Shift + P** to open the command palette, type `RAB: Initialize Workspace`, and hit **Enter**.

The VS Code extension now initializes the workspace, and creates a few directories and template files in the workspace. The directories and files created are displayed in the Explorer view on the left. The following directories and files are created:

- **api:** You can save all OpenAPI files in this folder.

 **Note:**

For the VS Code extension to package an OpenAPI document into an adapter bundle, rename the OpenAPI document as `openapi.resource.json`.

- **definitions:** The extension creates a default adapter definition document, `main.add.json`, in this folder. You can overwrite the default document with the data from a Postman collection or an OpenAPI file. Additionally, all the other adapter definition documents you generate are saved in this folder. The file names of all adapter definition documents are appended with `.add.json`.
 - **misc:** You can save all Postman collections in this folder. Ensure that the file names of all Postman collections are appended with `.postman_collection.json`.
 - **logo.svg:** A default logo file for your adapter is created under the root folder. You can replace this file with the required image.
2. Configure the publisher profile for the extension.
- a. In VS Code, click **OIC Rapid Adapter Builder** on the sidebar.
 - b. In the resulting view, click **Edit** in the Publisher Profiles section.
The `publisher-profiles.yaml` file opens in the editor.
 - c. Update the file with the client credentials you obtained earlier. See [Create a Client Application and Obtain Credentials](#). In addition, update the `active` field with the name of the profile to use.
 - d. Press **Ctrl+S** on your keyboard to save the file.

If you enter credentials for more than one Oracle Integration instance, the `.yaml` file stores them in a hierarchical structure, implemented by indenting the details.

The following example shows the details of two Oracle Integration instances, named `Test_Lab1` and `Test_Lab2`, stored in a `.yaml` file.

```
# select which profile to use.
active: Test_Lab1

# multiple profiles can be defined.
profiles:
  - name: Test_Lab1
    host: https://<oicinstance.oraclecloud.com>
    integrationInstance: example-instance1
    auth:
      tokenUrl: https://<idcs-token>/oauth2/v1/token
      clientId: ef8xxxxbf0c4101a4f74ae5fe301exx
      clientSecret: 0ebdae12-xxxx-4182-9051-8dbd83b2xxxx
      scope: https://
08DADxxxx9D4496F9FBAFxxCFA1B0657.integration.region-1.oraclecloud.co
m:443urn:opc:resource:consumer::all
```

```
- name: Test_Lab2
  host: https://<oicinstance.oraclecloud.com>
  integrationInstance: example-instance2
  auth:
    tokenUrl: https://<idcs-token>/oauth2/v1/token
    clientId: ef9xxxxxbf0c4101a4f74ae5fe301exx
    clientSecret: 1ebdae12-xxxx-4182-9051-8dbd83b2xxxx
    scope: https://
18DADxxxx9D4496F9FBAFxxCFA1B0657.integration.region-2.oraclecloud.co
m:443urn:opc:resource:consumer::all
```

Construct the API Request for OpenWeather

To make an API request for any application, you must identify the request format using the API documentation available for the application.

For the OpenWeather application, the weather-request API has the following format:

```
https://api.openweathermap.org/data/2.5/weather?q={City Name}&appid={API Key}
```

As part of the request, you must specify the name of the location and your API key created in the OpenWeather application.

Obtain your OpenWeather API key and construct the request.

1. Create an account on OpenWeather to get your API key. Visit the [OpenWeather Sign Up](#) page.

You'll receive an API key on your registered email address after you've verified your email address and completed the sign up process. Note that the API key is activated a few hours after the registration.

2. Construct your API request for a specific location. Here is an example API call for London city with an example API key:

```
api.openweathermap.org/data/2.5/weather?
q=London&APPID=09d2d589238d5f0d858f6ea46f5d71bf
```

Create a Postman Collection Using the API Request

Create a collection in the Postman application, add and test the API request you just constructed, and export the collection to a JSON file.


1. Open the Postman application and sign in with your account.
2. Create a collection.
 - a. With **Collections** selected in the left pane, click **New**, and in the resulting dialog, click **Collection**.
A new, blank collection is created and listed in the left pane.
 - b. Under the **Overview** tab on the right, click the default collection name (**New Collection**) to rename the collection. Enter `Get Weather Data Collection` as the new name.

3. Add the API request you constructed earlier to the collection, test it, and save the response as an example.
 - a. In the left pane, click **Add a request** under the newly-created collection.
A new, blank request is created and displayed.
 - b. In the **URL** field of the request, with the GET command selected, enter the API request you constructed earlier. Note that the query parameters `q` and `appid` are automatically populated in the table according to the data in your API request.
 - c. Click **Send** to test the request.

The Response section displays the weather data received. Here's an example response:

```
{
  "coord": {
    "lon": -0.1257,
    "lat": 51.5085
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 279.09,
    "feels_like": 276.39,
    "temp_min": 277.86,
    "temp_max": 280.26,
    "pressure": 1015,
    "humidity": 86
  },
  "visibility": 10000,
  "wind": {
    "speed": 3.6,
    "deg": 350
  },
  "clouds": {
    "all": 75
  },
  "dt": 1704540946,
  "sys": {
    "type": 2,
    "id": 2075535,
    "country": "GB",
    "sunrise": 1704528308,
    "sunset": 1704557195
  },
}
```

```
"timezone": 0,  
"id": 2643743,  
"name": "London",  
"cod": 200  
}
```

- d. In the Response section, click **Save as example** to save the response.
A new example is created and displayed.
For the Rapid Adapter Builder to generate a schema in the adapter definition document, you must save the response received for your request.
 - e. In the left pane, click the request item to switch back to the Request page from the Example page, and then click **Save** to save the entire request.
4. Export the Postman collection to your workspace.
 - a. In the left pane, point to **Get Weather Data Collection**, click **View more actions** , and then click **Export** from the resulting menu.
 - b. In the Export collection dialog, leave the default options selected, and click **Export**.
 - c. In the Select path to save file window, navigate to the folder that you selected and initialized as workspace previously. See [Configure the VS Code Extension for Rapid Adapter Builder](#). In the workspace folder, open the **misc** directory, and click **Save** to export the file to this directory.


The Postman collection is saved as a JSON file, named `Get Weather Data Collection.postman_collection.json`.

 **Note:**

For the Rapid Adapter Builder to convert the collection into an adapter definition document, the file name of the Postman collection should always contain the suffix `.postman_collection`.

Generate the Adapter Definition Document

Using the VS Code extension for Rapid Adapter Builder, you can generate an adapter definition document from the exported Postman collection.

1. In VS Code, click **Explorer**  on the sidebar to bring up the Explorer view.
The directory structure of your workspace folder is displayed.
2. Click the **misc** directory listing to expand it.
3. Right-click the exported collection file (`Get Weather Data Collection.postman_collection.json`), and then click **RAB: Convert Postman Collection**.
4. In the Select request(s) pane, select the request to convert, and click **Done**.
5. In the dialog box that appears, click **Update main**.

Now, the VS Code extension converts the data from the Postman collection and adds it to the default adapter definition document (`main.add.json`) within the `definitions` directory of your workspace. In addition, the document opens in the VS Code editor.

6. In the editor, update the `info` section of the document as follows:

```
"id": "oracledemo:getweather",
  "displayName": "Get Weather Data",
  "description": "Get Weather Data Connector",
```

When you publish the adapter to Oracle Integration, the display name you specify here appears as the adapter's name.

7. Ensure that the adapter definition document contains an API response schema.
8. Press **Ctrl+S** on your keyboard to save the file.


Register the Adapter on Oracle Integration


After you have validated the adapter definition document, you can make your adapter available on Oracle Integration.

1. Press **Ctrl + Shift + P** to open the command palette. From the commands displayed, click the `RAB: Register RAB Bundle` command.

Now, the VS Code extension registers the adapter on your Oracle Integration instance. The adapter is registered on the instance specified as *active* in the publisher profile.

After the registration is complete, a success message is displayed in the VS Code editor's bottom panel, on the **Output** tab.

2. Log in to your Oracle Integration instance and check for the registered adapter.
 - a. On the Oracle Integration Home page, click **Show/Hide navigation menu**  to view the navigation pane.
 - b. In the navigation pane, click **Design**, and then click **Adapters**.
 - c. On the Adapters page, click the **Search** icon, type the name of your adapter in the **Search** field, and hit **Enter**.

You'll see the adapter you published in the search results. Point to the adapter's row and click **View**  to view its details.


Congratulations! You've now successfully built and published your first adapter using the Rapid Adapter Builder.

Related Documentation

- [Using the Rapid Adapter Builder with Oracle Integration 3](#)
- [Getting Started with Oracle Integration 3](#)
- [Using Integrations in Oracle Integration 3](#)

Validate the Adapter Definition Document

Before you publish the adapter to Oracle Integration, you must validate the adapter definition document. Validating the document ensures that it does not contain any syntactic or semantic errors.

1. In VS Code, click **Explorer**  on the sidebar.
The directory structure of your workspace folder is displayed.
2. Click the **definitions** directory listing to expand it.
3. Right-click the adapter definition document (`main.add.json`), and then click **RAB: Validate**.

Now, the VS Code extension validates the document. If your document is valid, a success message is displayed in the VS Code editor's bottom panel, on the **Output** tab.

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.

Oracle Cloud Get Started with the Rapid Adapter Builder in Oracle Integration 3,
F85991-03

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

Documentation that describes how to quickly build an adapter using the Rapid Adapter Builder platform in Oracle Integration.