Quick Start Guide

Welcome to the WSO2 Micro Integrator Quick Start Guide, your step-by-step tutorial for getting started with WSO2 Micro Integrator (MI). WSO2 MI enables you to build, deploy, and manage integration solutions with ease, providing flexibility and scalability to connect applications, services, and systems.

In this guide, you'll learn the basics of setting up and using WSO2 MI to create and deploy a basic integration flow with minimal hassle.

Prerequisites

The following software and configurations are required to proceed with this tutorial:

- **Java Development Kit (JDK):** Version 11 or 17 is required. Set up a compatible JDK. Ensure the JDK is properly configured in your system's PATH environment variable.

  > ℹ️ **Info**
  >
  > For more information on setting the `JAVA_HOME` environment variable for different operating systems, see the Install and Setup documentation.

- **Apache Maven:** Ensure Apache Maven is installed (version 3.6.0 onwards) and its path is correctly set within the system's PATH environment variable.

  > ℹ️ **Info**
  >
  > For more information on installing Apache Maven, see the Apache Maven documentation.

- **WSO2 Micro Integrator 4.3.0 Runtime:** Set up WSO2 Micro Integrator 4.3.0 runtime on your machine.

    a. Download the Micro Integrator 4.3.0 distribution as a ZIP file from here.

    b. Extract the ZIP file. Hereafter, this extracted folder will be referred to as the `<MI_HOME>` folder.

- **Visual Studio Code (VS Code):** with the Micro Integrator for VS Code extension installed.

  > ℹ️ **Info**
  >
  > Follow the Install Micro Integrator for VS Code documentation for a complete installation guide.

After completing the step above, follow the instructions below to create your first integration solution:

What you'll build

Let's try a simple scenario where the client sends a request to a `HelloWorld` API deployed in the WSO2 Micro Integrator and the API calls a backend service and returns its response. The backend service responds a `Hello World!!!` message, and the API deployed in the WSO2 Micro Integrator forwards this response to the client.
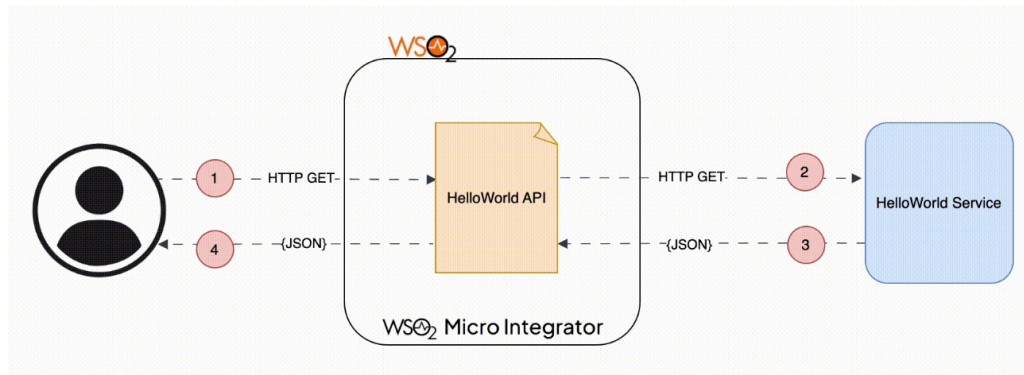

You can use the following HelloWorld service as the backend service.

URL	https://apis.wso2.com/zvdz/mi-qsg/v1.0
HTTP Method	GET

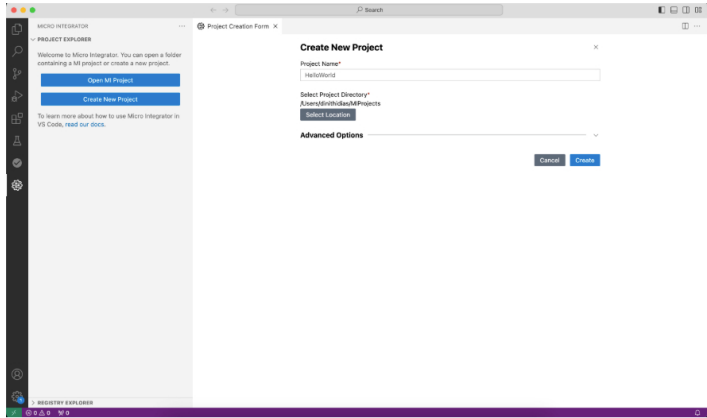## Step 1 - Create a new integration project

To develop the above scenario, let's get started with creating an integration project in the Micro Integrator extension installed VS Code.

1. Launch VS Code with the Micro Integrator extension installed.

2. Click on the Micro Integrator icon on the Activity Bar of the VS Code editor.
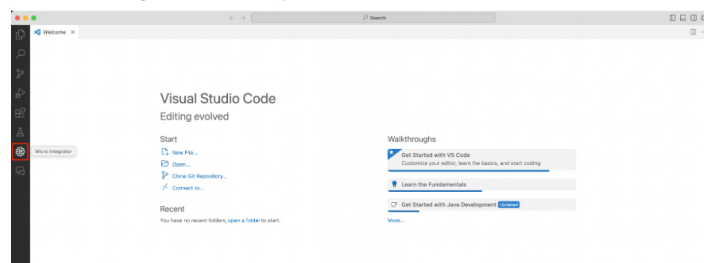
3. Click **Create New Project** on **Design View**. For more options to create a new integration project, see Create an Integration Project.

4. In the **Project Creation Form**, enter `HelloWorld` as the **Project Name**.

5. Provide a location under the **Select Project Directory**.
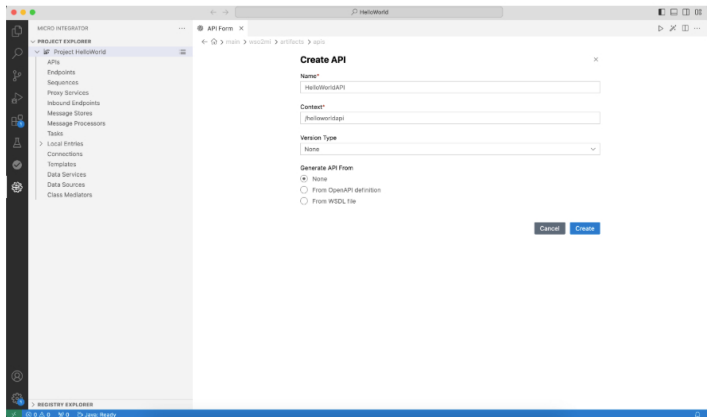

6. Click **Create**.

Step 2 - Create an API

Now the integration project is ready to add an API. In this scenario, the API calls a backend service and responds to the client. First, let's create an API.

1. Go to **Micro Integrator Project Explorer** > **APIs.**

2. Hover over **APIs** and click the **+** icon that appears to open the **Create API** form.

3. Enter `HelloWorldAPI` as the API **Name**. The API **Context** field will be automatically populated with the same value.

4. Click **Create**.

Once you create the API, a default resource will be automatically generated. You'll use this resource in this tutorial. To learn how to add a new resource to an API, see the Add new resource documentation.

## Step 3 - Design the integration

Now it is time to design your API. This is the underlying logic that is executed behind the scenes when an API request is made. In this scenario first, you need to call the backend service. For that, you have to add an endpoint.

1. Navigate to the **MI Project Explorer** > **Endpoints**.

2. Hover over Endpoints and click the + icon that appears.

3. Select **HTTP Endpoint** from the **Create Endpoint** interface.

4. Specify the following values to create the HTTP endpoint for the backend service.

Parameter	Value
Endpoint Name	`HelloWorldEp`
URI Template	`https://apis.wso2.com/zvdz/mi-qsg/v1.0`
HTTP Method	`GET`

5. Click **Create**.

Now you have to add a Call Mediator to call the backend service.

6. Open the **Resource View** of the API resource.

a. Go to **MI Project Explorer** > **APIs**.

b. Under `HelloWorldAPI`, click the default API resource to open the **Resource View** of the API resource.

7. Click on the **+** icon to open the mediator palette.

8. Select **Call Endpoint** mediator under **Mediators** > **Generic**.

9. Under **Endpoint**, select the created `HelloWorldEp` endpoint from the dropdown.

10. Click **Submit**.

    Now let's add a Respond Mediator to respond the message to the client.

11. Click on the **+** icon placed just after the Call mediator to open the mediator palette.

12. Select **Respond** mediator under **Mediators** > **Generic**.

13. Click **Submit**.
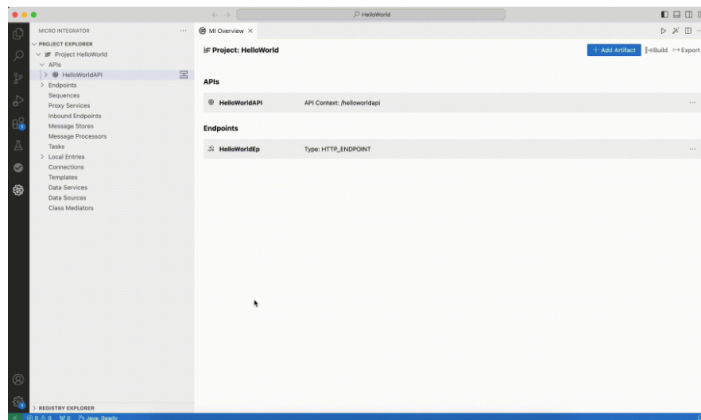
Following is what you'll see in the **Source View** of the VS Code.
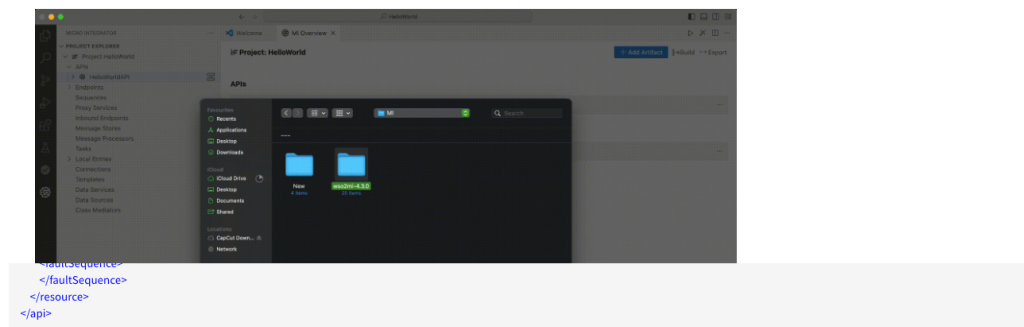
> ℹ️ **Info**
>
> You can view the source view by clicking on the **Show Source** (`</>`) icon located in the top right corner of the VS Code.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<api context="/helloworldapi" name="HelloWorldAPI" xmlns="http://ws.apache.org/ns/synapse">
  <resource methods="GET" uri-template="/">
    <inSequence>
      <call>
        <endpoint key="HelloWorldEp"/>
      </call>
      <respond/>
    </inSequence>
    <faultSequence>
    </faultSequence>
  </resource>
</api>
```

Step 4 - Add MI server to run integration

You need to configure the downloaded and extracted WSO2 MI server in the Micro Integrator extension installed VS Code to run the integration solution. Let's proceed with the following steps.

1. Open the VS Code **Command Palette** by selecting **View** > **Command Palette** from the menu, or by using the shortcut `Command + Shift + P` on macOS or `Ctrl + Shift + P` on Windows.

2. Select **MI: Add MI server** from the list of available commands.

3. Click **Add MI server** to add a Micro Integrator server.

4. Select the folder where `<MI_HOME>` is located. This will be set as the **current server path**.


```
    </faultSequence>
  </resource>
</api>
```

Congratulations! Now, you have created your first integration service.

Additionally, you can use the Integration Control Plane (ICP) to observe details of the deployed artifacts.

What's next?

Try more tutorials and examples.