**Practical Exercise: Getting Started with WSO2 Integration Studio**

Training Objective

Understand the functionality of WSO2 Integration Studio.

High Level Steps

* Install WSO2 Micro Integrator.
* Overview of WSO2 Integration Studio.

Detailed Instructions

**Installing WSO2 Integration Studio**

WSO2 Integration Studio provides a comprehensive development experience for building integration solutions.

Installation prerequisites

|  |  |
| --- | --- |
| **Processor** | Intel Core i5 or equivalent |
| **RAM** | 4 GB minimum, 8 GB recommended |
| **Disk Space** | Approximately 4 GB |

Install and run WSO2 Integration Studio

Follow the steps given below.

1. Go to the [API Manager Tooling web page](https://wso2.com/api-management/tooling/), and download WSO2 Integration Studio.

**Note**

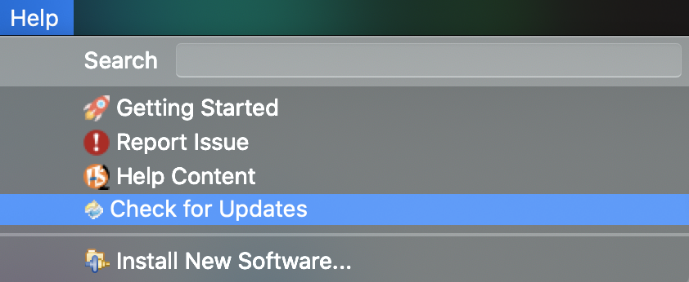
* + If you are a MacOS user, be sure to add it to the **Applications** directory.
  + If you are a Microsoft Windows user, extract it outside the **Programs** directory. This is done because the Integration Studio requires permission to write to files.

1. Run the **Integration Studio** application to start the tool.

Get the latest updates

If you have already installed and set up WSO2 Integration Studio, you can get the latest updates as follows:

1. Open WSO2 Integration Studio on your computer.
2. Go to **Help** -> **Check for Updates**.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/get-tooling-updates.png)

1. Once the update check is completed, you can select all the available updates and install.

**Checking the version**

You can check the version of the Integration Studio as below.

* For MacOS : Integration Studio > About Integration Studio
* For Windows/Linux : Help > About Integration Studio

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/about-integration-studio.jpg)

Troubleshooting

If you get an error message about the file being damaged or that you cannot open the file when you try to start the tool on a MacOS, change the MacOS security settings as described below.

1. Go to **System Preferences** -> **Security & Privacy** -> **General**.
2. Under **Allow apps downloaded from**, click **Anywhere** .
3. Thereafter, select **IntegrationStudio** from the **Applications** menu in your Mac.

# Quick Tour - WSO2 Integration Studio

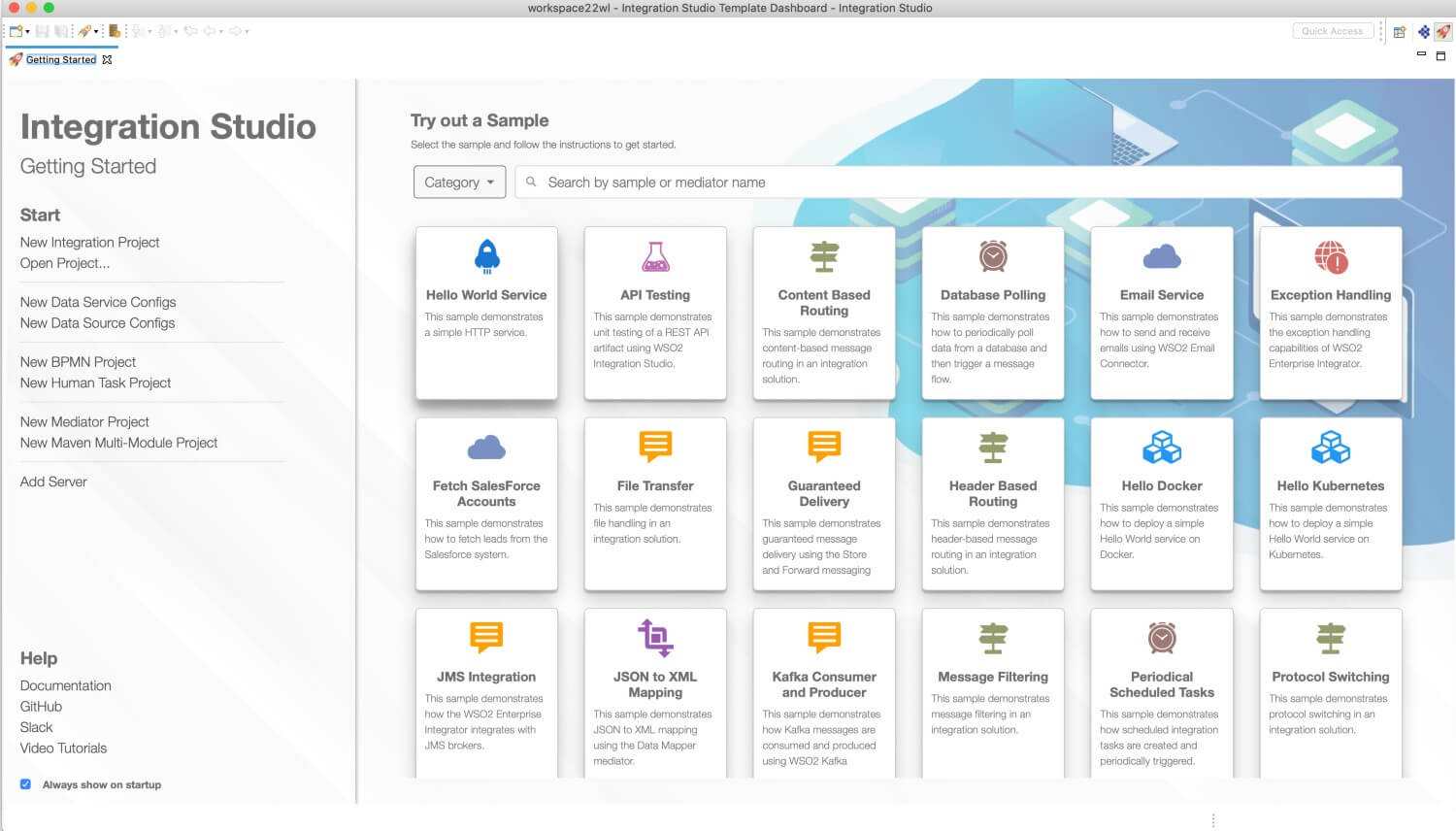
WSO2 Integration Studio is your development environment for designing, developing, debugging, and testing integration solutions. As an integration developer, you can execute all the phases of the development lifecycle using this tool. When your integration solutions are production-ready, you can easily push the artifacts to your continuous integration/continuous deployment pipeline.

**Tip**

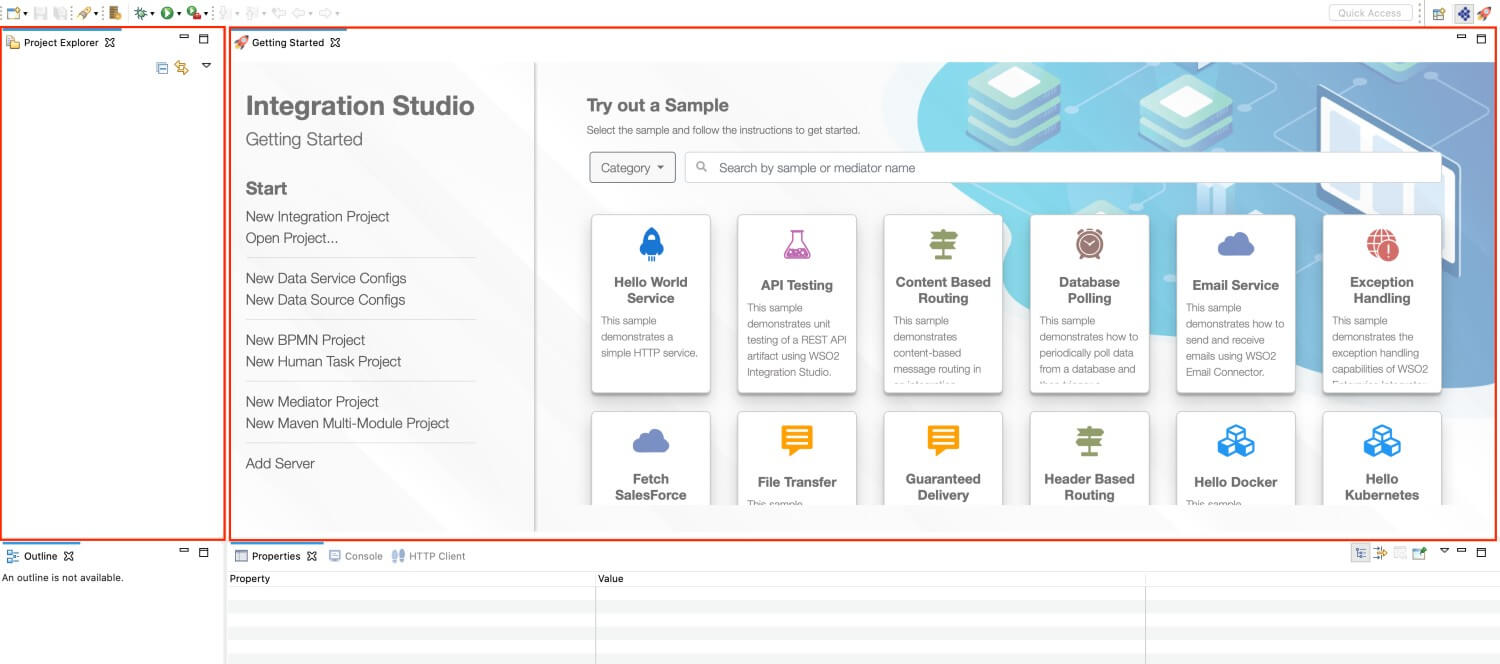
The base of the WSO2 Integration Studio is Eclipse IDE. You can install any supported Eclipse plugin for Integration Studio by navigating to **Help** -> **Install New Software**.

## **Getting Started**

When you open WSO2 Integration Studio, you will see the **Getting Started** view in the tool's workbench.



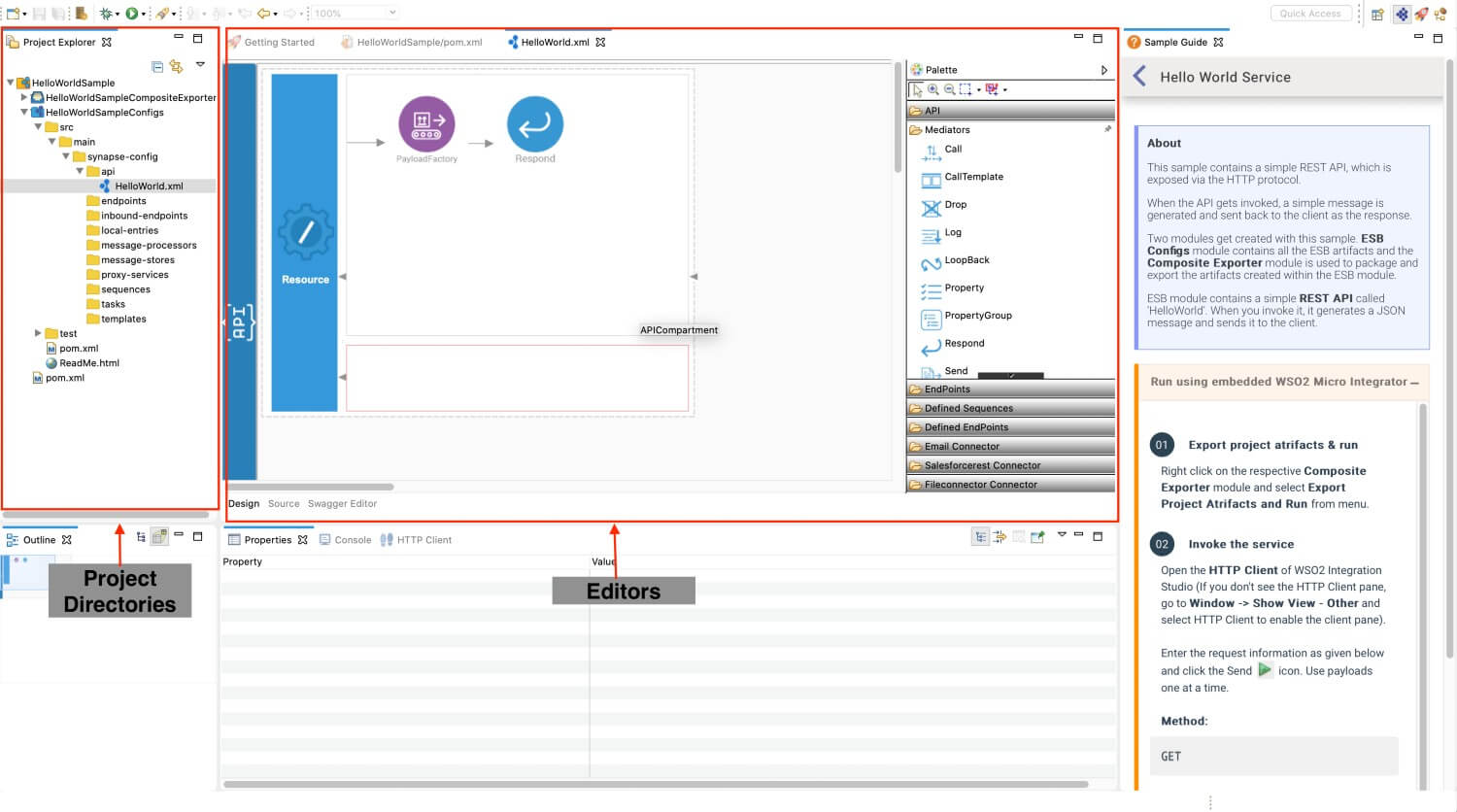
You can also click the https://apim.docs.wso2.com/en/4.0.0/assets/img/integrate/workbench/icon-getting-started-workbench.png icon at the top-right of the workbench to open the **Project Explorer** alongside the **Getting Started** tab as shown below.



To get started, you need to first create the required project directories. Alternatively, you can use an integration sample, which will generate the required projects and files for a specific use case.

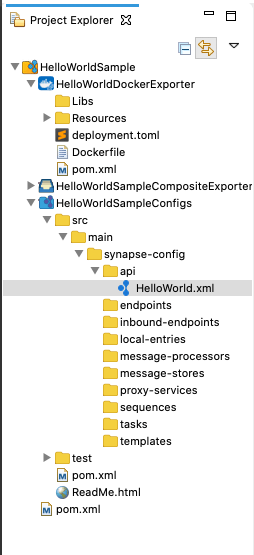
|  |  |
| --- | --- |
| Project Directories | Use the links on the **Getting Started** view to create the required projects. These project directories are saved to your workspace and they can later be accessed from the [Project Explorer](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/wso2-integration-studio/#project-explorer) view of WSO2 Integration Studio.  https://apim.docs.wso2.com/en/4.0.0/assets/img/integrate/workbench/getting-started-view.jpg |
| Samples | The **Getting Started** view lists a set of sample projects and integration artifacts that represent common integration scenarios. You can use these to explore WSO2 Micro Integrator and to try out common integration use cases. The [sample guide](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/wso2-integration-studio/#sample-guide) will provide instructions on how to run the samples.  https://apim.docs.wso2.com/en/4.0.0/assets/img/integrate/workbench/getting-started-templates.jpg |
| Sample Guide | The sample guide is a **Help** pane, which provides documentation on how to use the [integration sample scenarios](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/wso2-integration-studio/#templates). You can follow the instructions given in the guides to deploy and test each sample scenario.  https://apim.docs.wso2.com/en/4.0.0/assets/img/integrate/workbench/workbench-template-guide.png |

Once you have created the required set of projects and artifacts, you can start working with the project directories and artifact editors shown below.



## **Project Explorer**

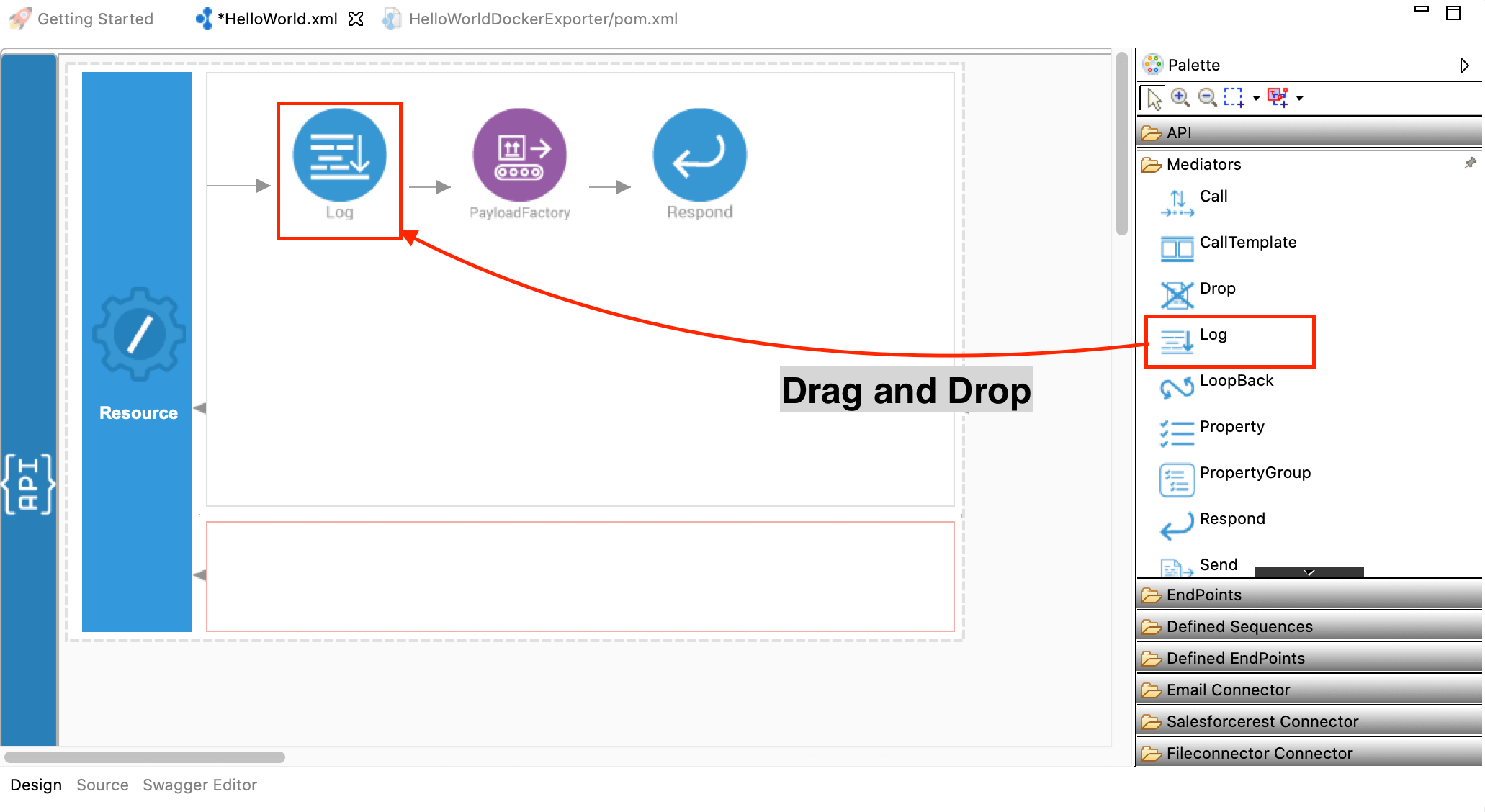
The project explorer provides a view of all the project directories created for your integration solution. Shown below is the project explorer of a working project.



## **Graphical Editor**

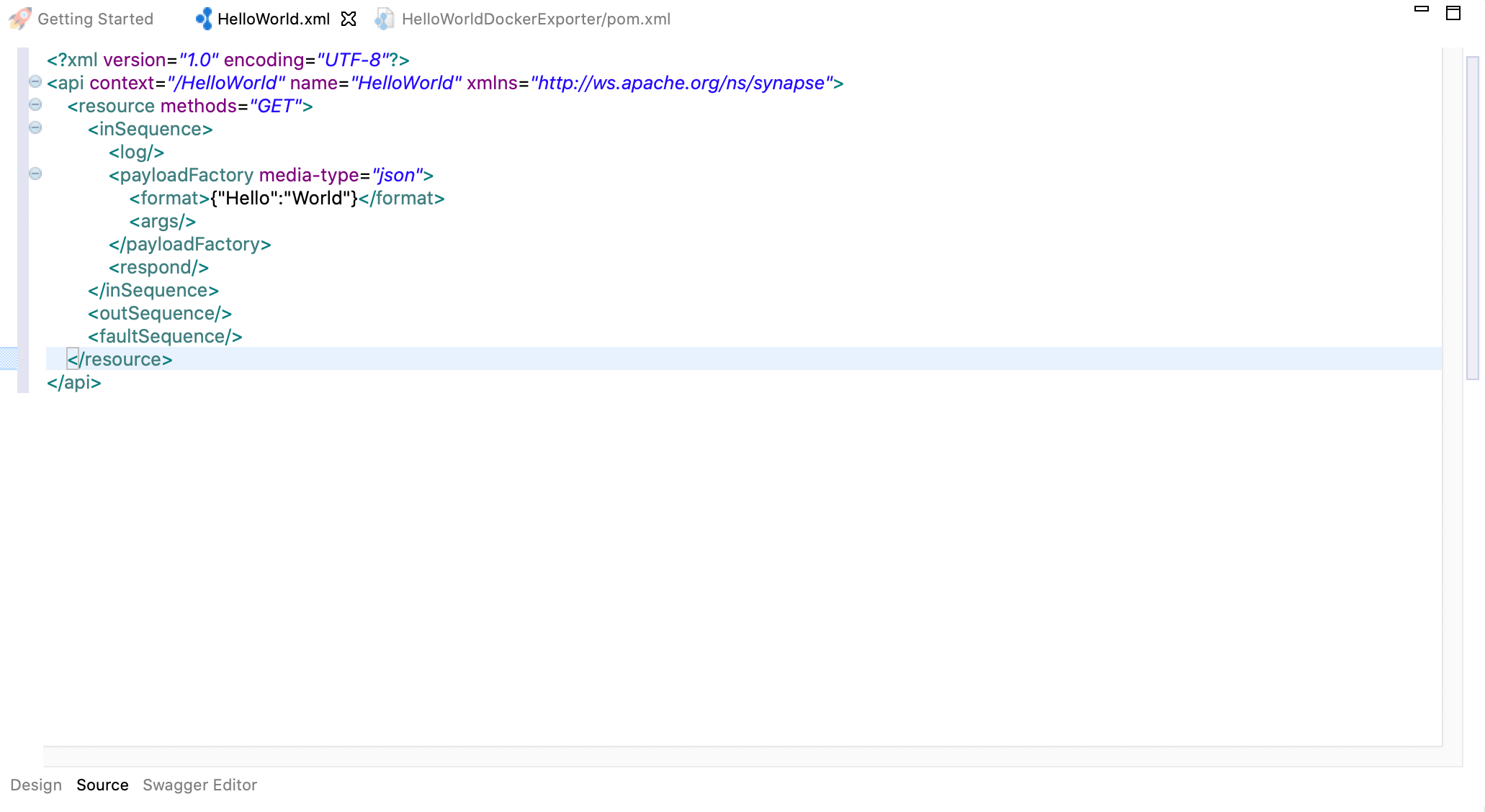
The graphical editor of WSO2 Integration Studio is a drag-and-drop editor for designing integration workflows. To access the graphical editor, you must first create a REST API, Proxy Service, Inbound Endpoint, or Sequence artifact.

Once you open the graphical editor, the **Palette** to your left lists all the integration artifacts that you can use. You can drag the required artifacts to the canvas on your right and design your integration flow. The parameters for each artifact can be configured using the [Properties](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/wso2-integration-studio/#properties) view.



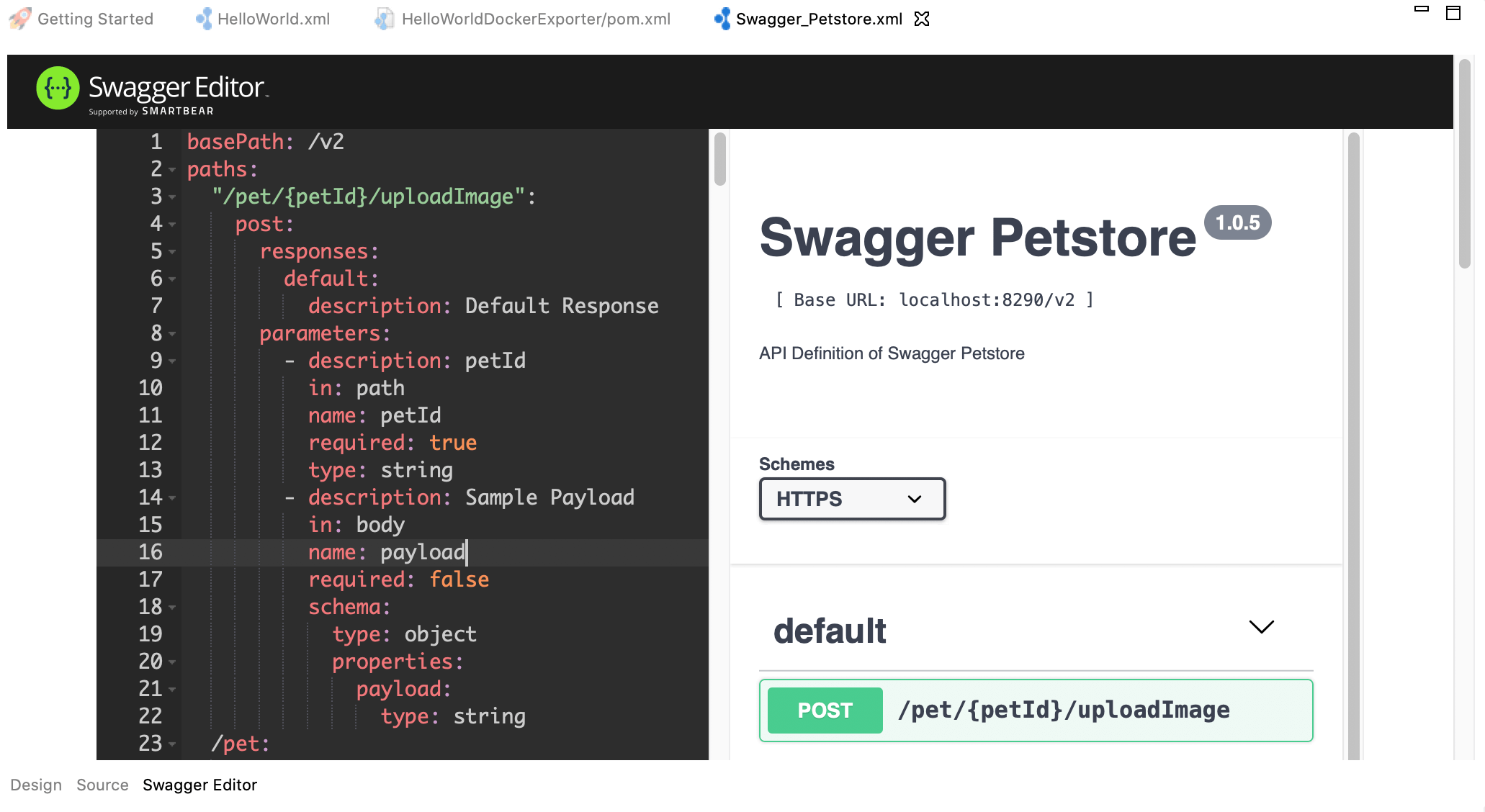
## **Source Editor**

When you open any integration artifact from the project explorer, you will have a source editor in addition to the graphical editor. This editor allows you to write or edit your integration solution using the source code (synapse).



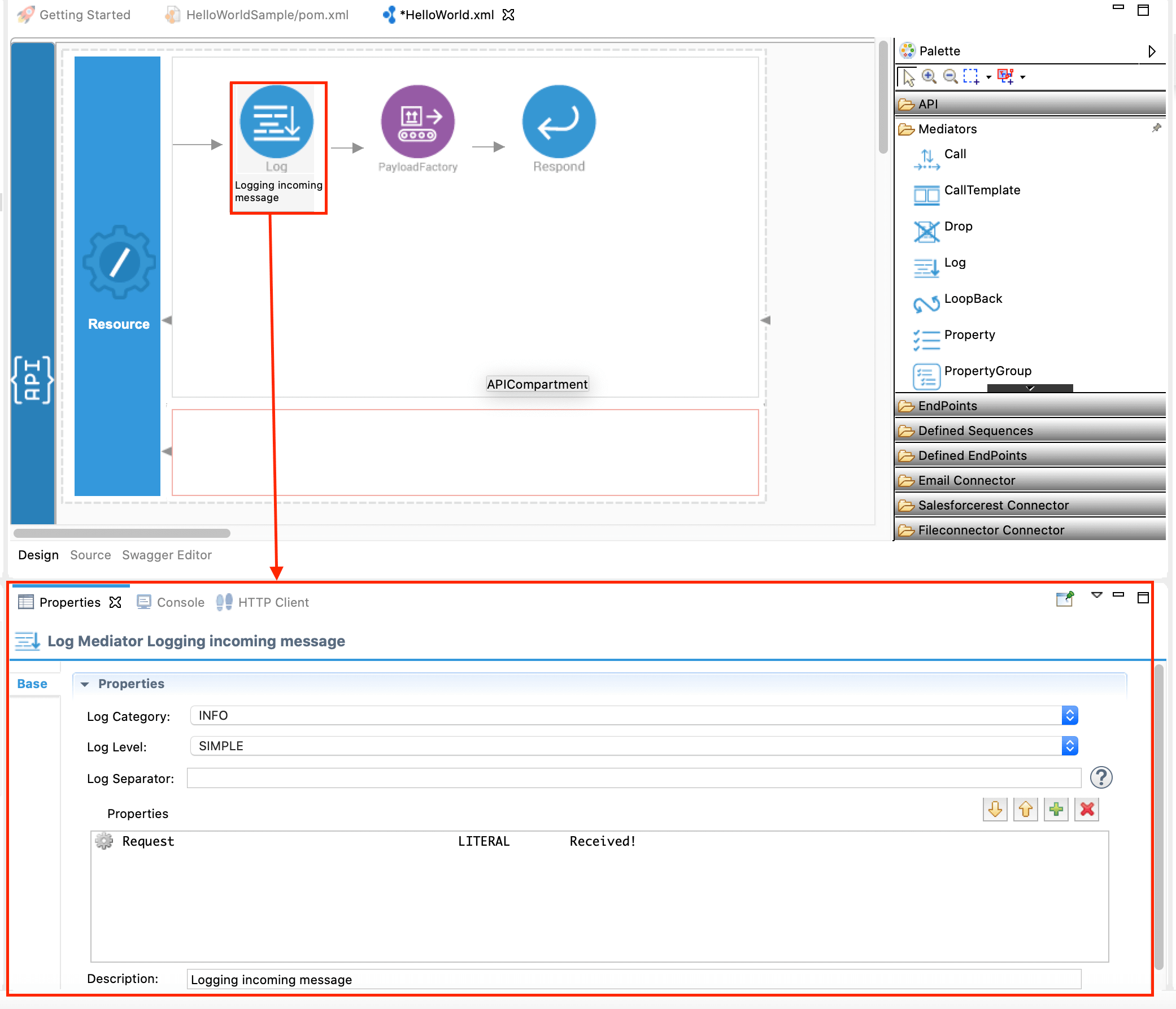
## **Swagger Editor**

The swagger editor is available when you create a REST API. This is in addition to the graphical editor and the source editor. The swagger editor can be used to write or edit your integration solution using the swagger definition. The Swagger UI allows you to visualize and interact with the REST API.



## **Properties**

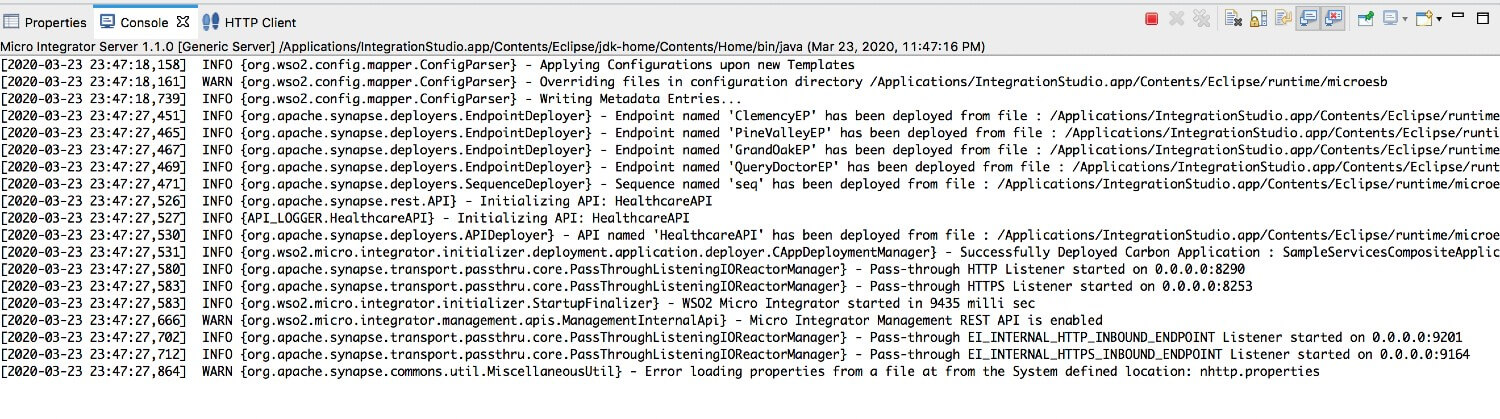
The properties view allows you to configure the properties and parameters that define the integration artifacts in your integration flow. When you double-click an artifact in the graphical editor, the **Properties** view for that artifact will open. Alternatively, you can right-click the artifact and click **Show Properties** to open this view.



## **Console**

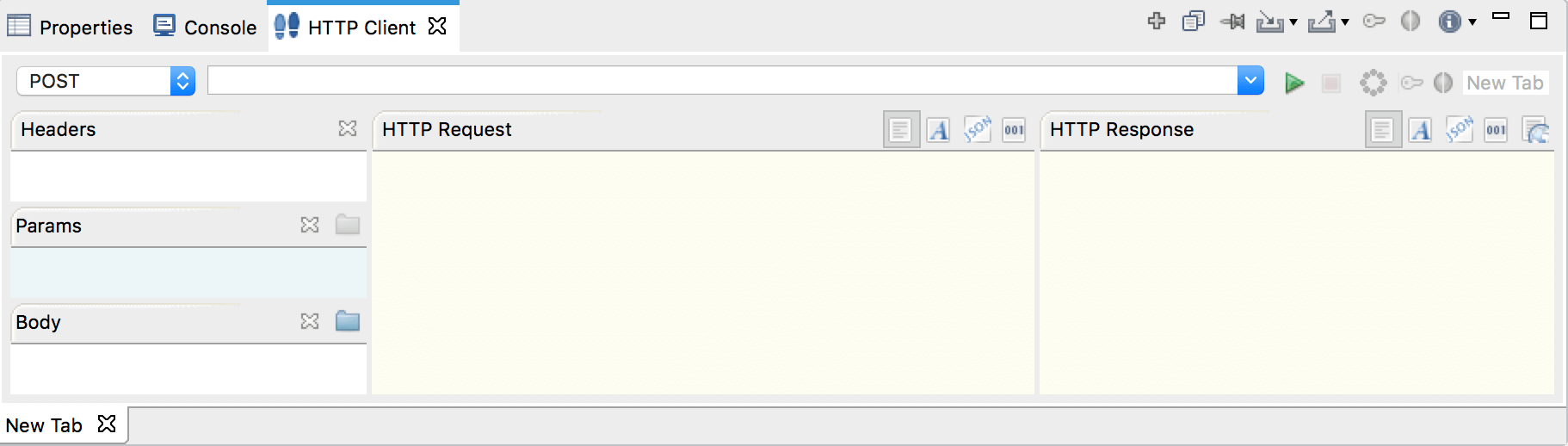
The Console view displays a variety of console types depending on the type of development and the current set of user settings. The three consoles that are provided by default with WSO2 Integration Studio are:

* **Process Console**: Shows standard output, error, and input.
* **Stacktrace Console**: Well-formatted Java stack trace with hyperlinks to specific source code locations.
* **CVS Console**: Displays output from CVS operations.



## **HTTP Client**

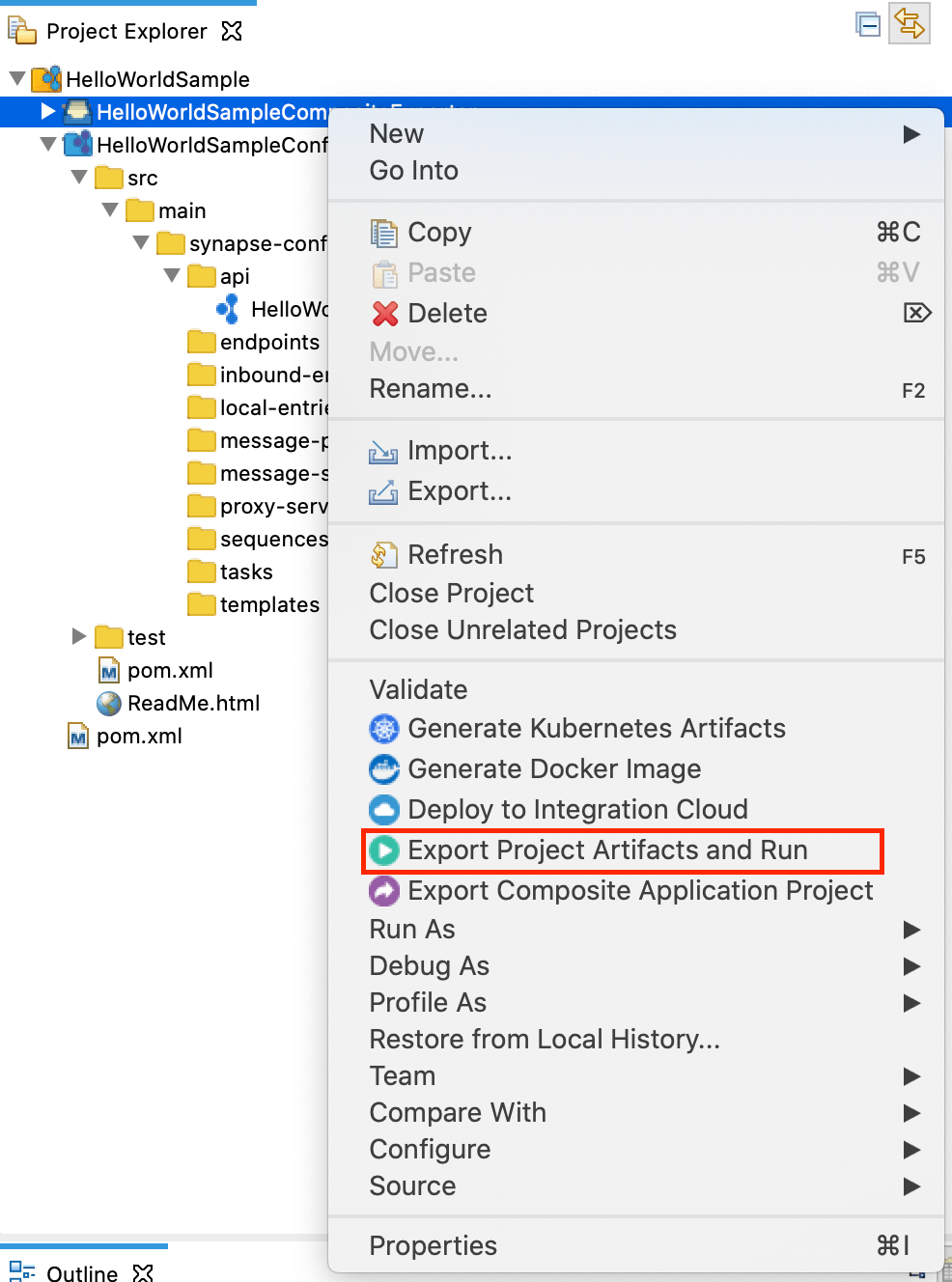
An embedded RESTful HTTP client is shipped with WSO2 Integration Studio to support testing. Once you have deployed your integration solutions in the Micro Integrator server, you can invoke the solutions using this client within WSO2 Integration Studio.



## **Embedded Micro Integrator**

WSO2 Integration Studio is shipped with an embedded Micro Integrator server, which allows developers to deploy and run integration artifacts during the development phase. To deploy the artifacts and to run the embedded Micro Integrator, right-click the composite application project (which includes your artifacts) and click **Export Project Artifacts and Run**.

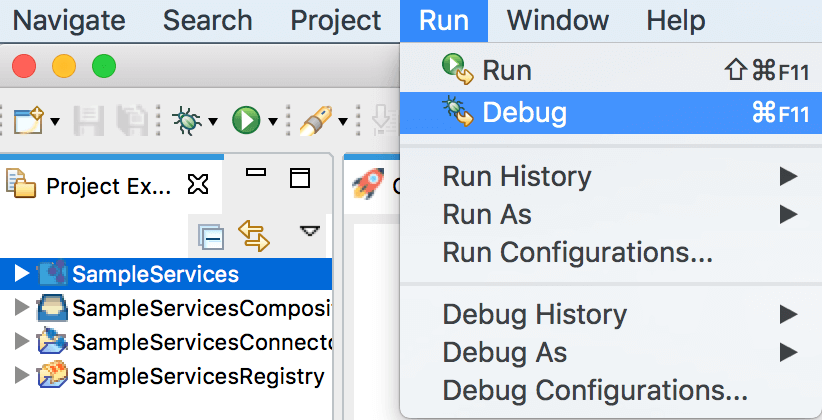
Find out more about [using the embedded Micro Integrator](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/using-embedded-micro-integrator).



## **Inbuilt Debugging Capabilities**

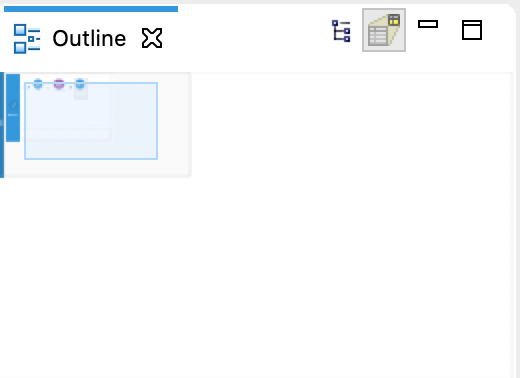
WSO2 Integration Studio is shipped with mediation debugging capabilities, which allows developers to debug an integration project using the tool. The embedded Micro Integrator server and debugging capabilities enable developers to comprehensively test, debug, and improve integration solutions before the artifacts are released to a production environment.

You need to select your integration project in the project explorer and go to **Run -> Debug** as shown below. Find out more about [mediation debugging](https://apim.docs.wso2.com/en/4.0.0/integrate/develop/debugging-mediation).



## **Outline**

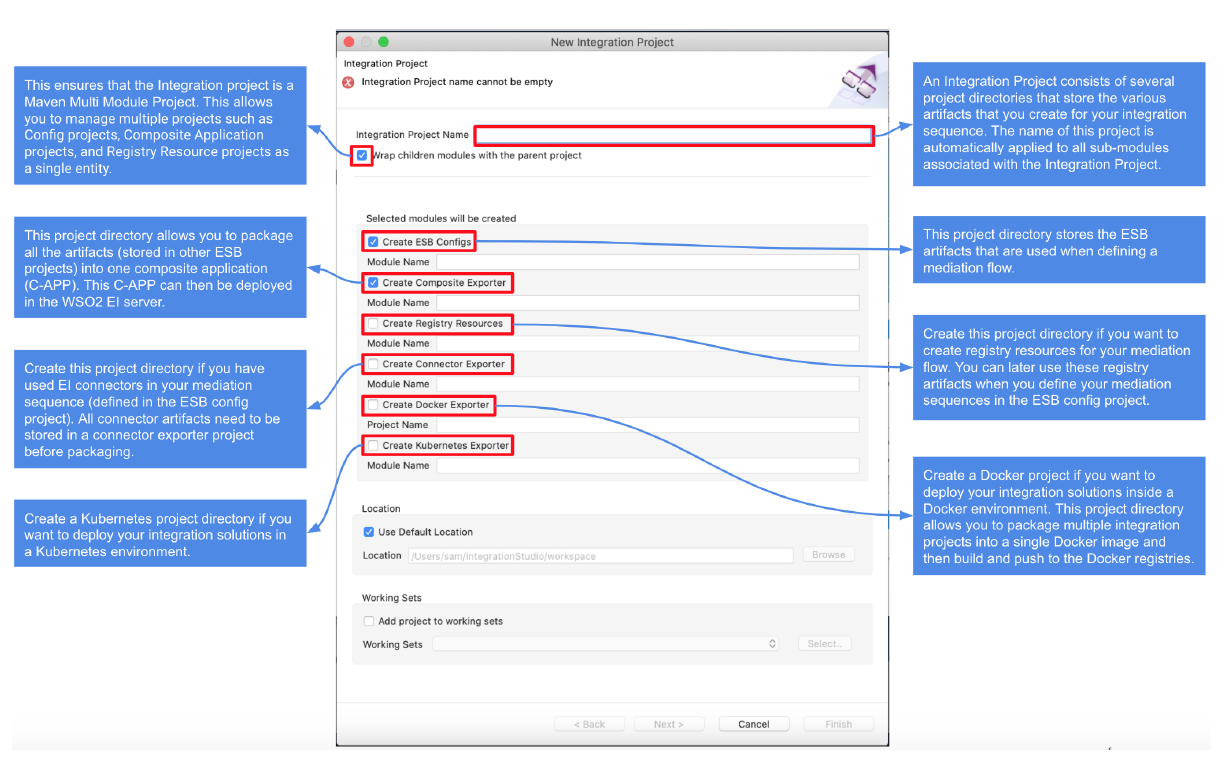
The Outline view displays an outline of a structured file that is currently open in the editor area and lists structural elements. It enables you to hide certain fields, methods, and types, and also allows you to sort and filter to find what you want. The contents of the outline are editor-specific. For example, in a Java source file, the structural elements are classes, fields, and methods. The contents of the toolbar are also editor-specific.



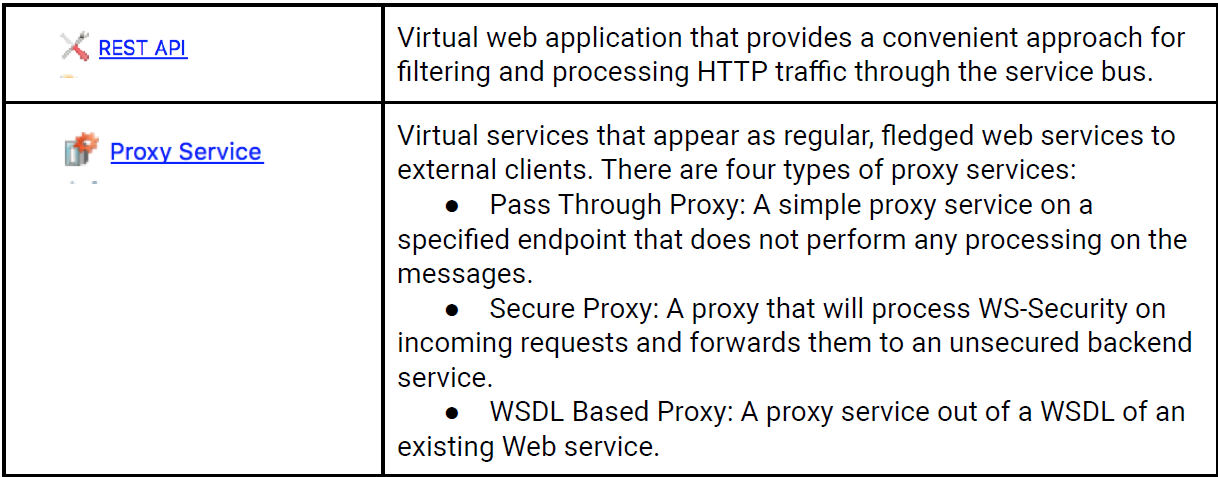
**Start with an Integration Project**

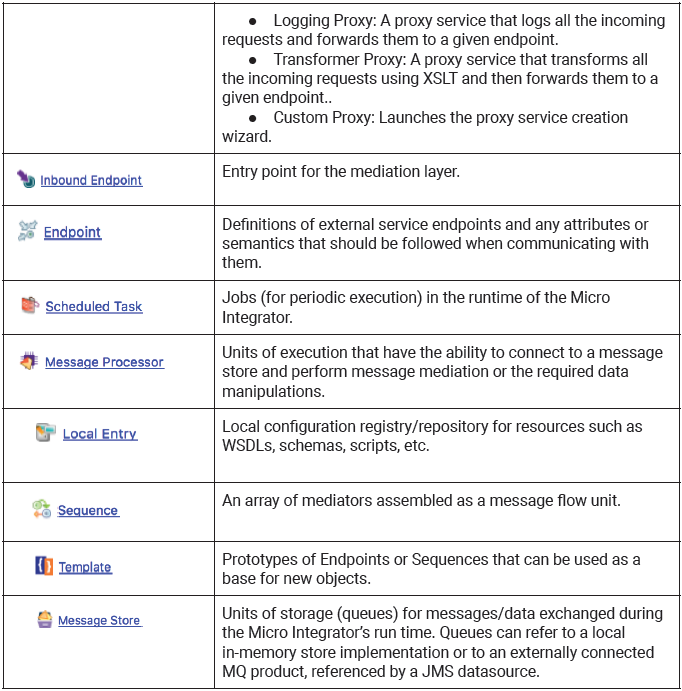
You can click **New Integration Project** to create a new maven multi module project that will

include all the sub projects for your integration solution.



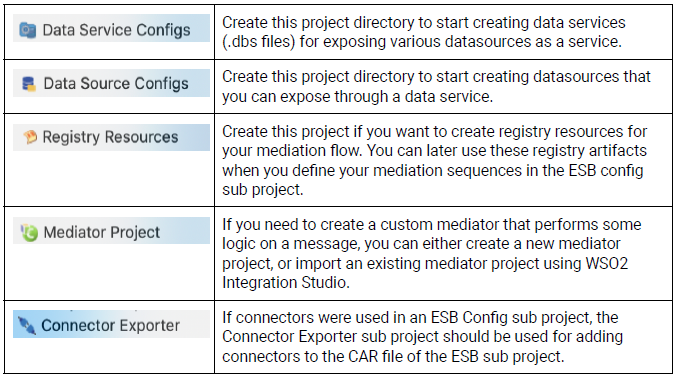
The **ESB Config** sub project in the integration project will contain the main integration artifacts:





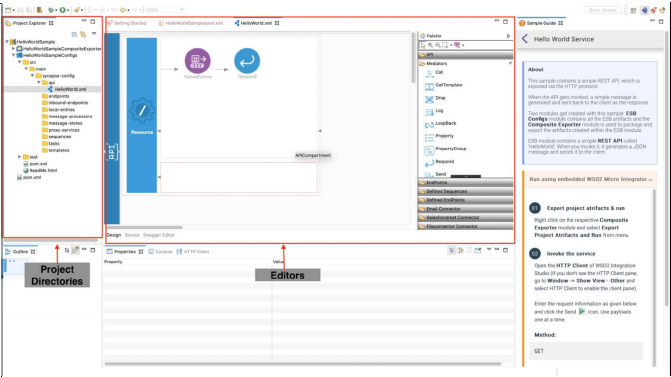
The following sub projects will contain the additional artifacts required for your integration

solution.



**Start with a Sample**

You can click a selected sample to create a new maven multi module project with the artifacts for that sample solution. Follow the instructions in the **Sample Guide** panel and run the sample.

****

**Best Practices**

* Always use an integration project (maven multi module project) as the parent project for your configuration artifacts. All other projects should be defined as sub projects under the parent project.
* Create a common project to have the main mediation logic and have all environment-dependent (dev/test/prod) artifacts in a separate project.