**Practical Exercise: Using Inbound Endpoints**

Training Objective

Configure an HTTP Inbound Endpoint using WSO2 Integration Studio.

You will expose the REST API you created in the previous lab exercise on a different port in the Micro Integrator using the HTTP Inbound API.

High-level Steps

* Create an HTTP Inbound Endpoint
* Test your configuration

Detailed Instructions

Using Inbound Endpoints

What you'll build

In this sample scenario, you will use an **Inbound Endpoint** to expose an already defined REST API through a different port. You can reuse the REST API that was defined in the [Sending a Simple Message to a Service](https://apim.docs.wso2.com/en/4.1.0/tutorials/integration-tutorials/sending-a-simple-message-to-a-service) tutorial. See [Creating an Inbound Endpoint](https://apim.docs.wso2.com/en/4.1.0/integrate/develop/creating-artifacts/creating-an-inbound-endpoint) for details on how to work with inbound endpoints using WSO2 Integration Studio.

Let's get started!

Step 1: Set up the workspace

Set up WSO2 Integration Studio as follows:

1. Download the relevant [WSO2 Integration Studio](https://wso2.com/api-management/tooling/) based on your operating system.
2. Set up the project from the [Sending a Simple Message to a Service](https://apim.docs.wso2.com/en/latest/tutorials/integration-tutorials/using-inbound-endpoints/sending-a-simple-message-to-a-service) tutorial:

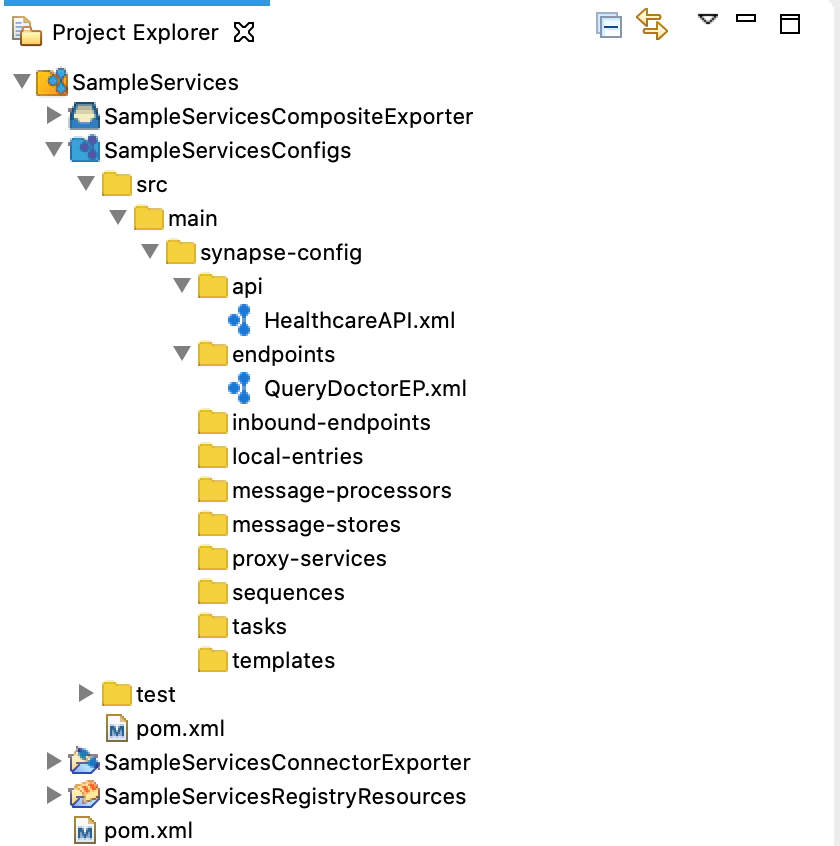
**Note**

This tutorial is a continuation of the [Sending a Simple Message to a Service](https://apim.docs.wso2.com/en/4.1.0/tutorials/integration-tutorials/sending-a-simple-message-to-a-service/) tutorial.

* 1. Download the [prepackaged project](https://github.com/wso2-docs/WSO2_EI/blob/master/Integration-Tutorial-Artifacts/Integration-Tutorial-Artifacts-EI7.1.0/sending-simple-message-tutorial.zip" \t "_blank).
  2. Open WSO2 Integration Studio and go to **File -> Import**.
  3. Select **Existing WSO2 Projects into workspace** under the **WSO2** category, click **Next**, and then upload the **prepackaged project**.

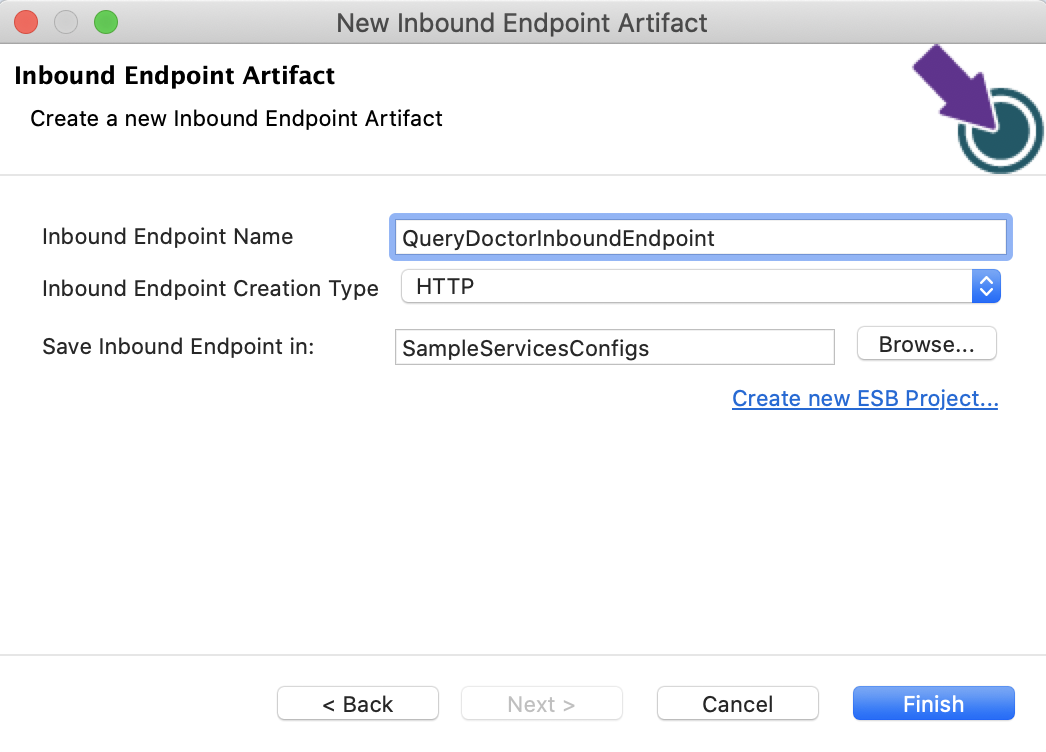
Step 2: Develop the Inbound Endpoint

1. Once you have exported the integration project as described in above, the project directory will appear with the artifacts as shown below. Note the 'HealthcareAPI' that is already included.



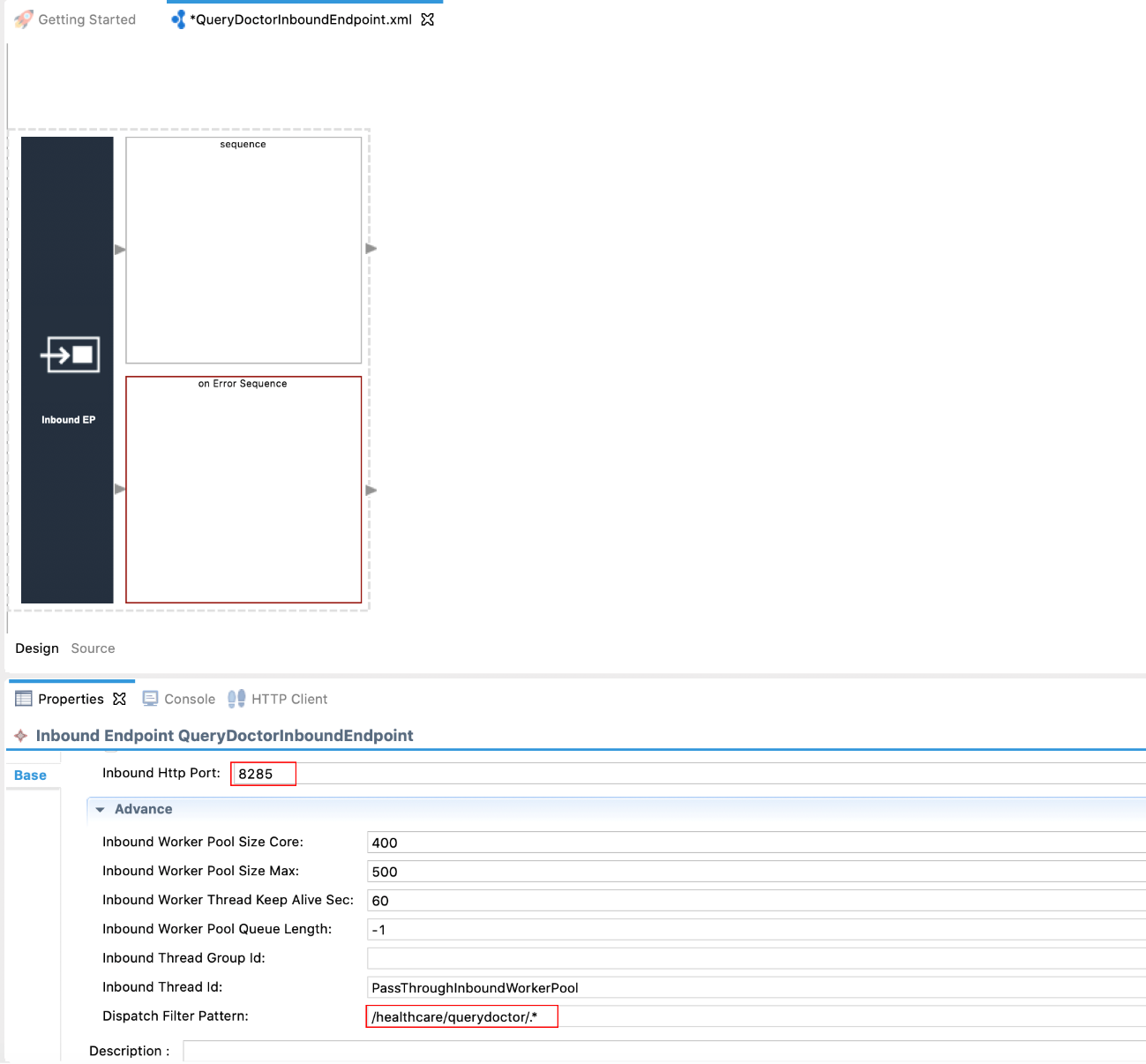
1. Right-click on **SampleServicesConfigs** and navigate to **New -> Inbound Endpoint**. Select **Create A New Inbound Endpoint** and click **Next**.
2. Enter the following details and click **Finish**.

|  |  |
| --- | --- |
| Parameter | Description |
| Inbound Endpoint Name | QueryDoctorInboundEndpoint |
| Inbound Endpoint Creation Type | HTTP |



1. Go to the **Properties** tab in the **Design** view and enter the following:

|  |  |
| --- | --- |
| Parameter | Description |
| Inbound HTTP port | 8285 |
| Dispatch Filter Pattern | /healthcare/querydoctor/.\* |



The endpoint will now get mapped to any URL that matches the above pattern provided. You will be exposing the health care API on a new port through this inbound endpoint.

Step 3: Package the artifacts

Package the artifacts in your composite exporter module (SampleServicesCompositeExporter) to be able to deploy the artifacts in the server.

1. Open the pom.xml file in the composite exporter module.
2. Ensure that the following artifacts are selected in the POM file.
   * HealthcareAPI
   * QueryDoctorEP
   * QueryDoctorInboundEndpoint
3. Save the changes.

Step 4: Build and run the artifacts

To test the artifacts, deploy the [packaged artifacts](https://apim.docs.wso2.com/en/latest/tutorials/integration-tutorials/using-inbound-endpoints/#step-3-package-the-artifacts) in the embedded Micro Integrator:

1. Right-click the composite exporter module and click **Export Project Artifacts and Run**.
2. In the dialog box that opens, confirm that the required artifacts from the composite exporter module are selected.
3. Click **Finish**.

The artifacts will be deployed in the embedded Micro Integrator and the server will start.

* See the startup log in the **Console** tab.
* See the URLs of the deployed services and APIs in the **Runtime Services** tab.

Step 5: Test the use case

Let's test the use case by sending a simple client request that invokes the service.

**Start the backend service**

1. Download the JAR file of the back-end service from [here](https://github.com/wso2-docs/WSO2_EI/blob/master/Back-End-Service/Hospital-Service-JDK11-2.0.0.jar).
2. Open a terminal, navigate to the location where your saved the back-end service.
3. Execute the following command to start the service:

java -jar Hospital-Service-JDK11-2.0.0.jar

**Send the client request**

Let's send a message to the **healthcare** REST API (through the inbound endpoint) on port 8285. You can use the embedded **HTTP Client** of WSO2 Integration Studio as follows:

1. Open the Postman application. If you do not have the application, download it from here : [Postman](https://www.postman.com/downloads/)
2. Add the request information as given below and click the **Send** button.

|  |  |
| --- | --- |
| Method | GET |
| URL | http://localhost:8285/healthcare/querydoctor/surgery |

If you want to send the client request from your terminal:

1. Install and set up [cURL](https://curl.haxx.se/" \t "_blank) as your REST client.
2. Open a command line terminal and execute the following command:

curl -v http://localhost:8285/healthcare/querydoctor/surgery

You will get the response shown below. The inbound endpoint has successfully invoked the REST API, and further, the response received by the REST API has been routed back to the client through the inbound endpoint.

[{"name":"thomas collins","hospital":"grand oak community

hospital","category":"surgery","availability":"9.00 a.m - 11.00 a.m","fee":7000.0},

{"name":"anne clement","hospital":"clemency medical center","category":"surgery","availability":"8.00 a.m - 10.00 A.m","fee":12000.0},

{"name":"seth mears","hospital":"pine valley community hospital","category":"surgery","availability":"3.00 p.m - 5.00 p.m","fee":8000.0}]