Creating a New Connector

You can write a new connector for a specific requirement that cannot be addressed via any of the existing connectors that can be downloaded from the [connector store](https://store.wso2.com/store/pages/top-assets).

Follow the steps given below to write a new connector to integrate with the **Google Books** service. You can then use the connector inside a mediation sequence to connect with Google Books and get information.

Writing a new connector

Follow the steps given below to write the new connector.

Prerequisites

Download and install Apache Maven.

Step 1: Creating the Maven project template

We will use the [maven archetype](https://github.com/wso2-extensions/archetypes/tree/master/esb-connector-archetype) to generate the Maven project template and sample connector code.

1. Open a terminal, navigate to the directory on your machine where you want the new connector to be created, and run the following command:

mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate -DarchetypeGroupId=org.wso2.carbon.extension.archetype -DarchetypeArtifactId=org.wso2.carbon.extension.esb.connector-archetype -DarchetypeVersion=2.0.4 -DgroupId=org.wso2.carbon.esb.connector -DartifactId=org.wso2.carbon.esb.connector.googlebooks -Dversion=1.0.0 -DarchetypeRepository=http://maven.wso2.org/nexus/content/repositories/wso2-public/

2. When prompted, enter a name for the connector. For example, **googleBooks**.

3. When prompted for confirmation, enter **y**.

The org.wso2.carbon.esb.connector.googlebooks directory is now created with a directory structure consisting of a pom.xml file, src tree, and repository tree.

Step 2: Adding the new connector resources

Now, let's configure files in the org.wso2.carbon.esb.connector.googlebooks/src/main/resources directory:

1. Create a directory named **googlebooks\_volume** in the /src/main/resources directory.
2. Create a file named listVolume.xml with the following content in the **googlebooks\_volume** directory:

<**template** name="listVolume" xmlns="http://ws.apache.org/ns/synapse">

<**parameter** name="searchQuery" description="Full-text search query string." />

<**sequence**>

<**property** name="uri.var.searchQuery" expression="$func:searchQuery" />

<**call**>

<**endpoint**>

<**http** method="get" uri-template="https://www.googleapis.com/books/v1/volumes?q={uri.var.searchQuery}" />

</**endpoint**>

</**call**>

</**sequence**>

</**template**>

1. Create a file named component.xml in the **googlebooks\_volume** directory and add the following content:

<?xml version="1.0" encoding="UTF-8"?>

<**component** name="googlebooks\_volume" type="synapse/template">

<**subComponents**>

<**component** name="listVolume">

<**file**>listVolume.xml</**file**>

<**description**>Lists volumes that satisfy the given query.</**description**>

</**component**>

</**subComponents**>

</**component**>

1. Edit the connector.xml file in the src/main/resources directory and replace the contents with the following dependency:

<?xml version="1.0" encoding="UTF-8"?>

<**connector**>

<**component** name="googleBooks" package="org.wso2.carbon.connector" >

<**dependency** component="googlebooks\_volume"/>

<**description**>wso2 sample connector library</**description**>

</**component**>

</**connector**>

1. Create a folder named **icon** in the /src/main/resources directory and add two icons.

**Tip**

You can download icons from the following location: [icons](http://svn.wso2.org/repos/wso2/scratch/connectors/icons/)

You are now ready to build the connector.

Step 3: Building the connector

Open a terminal, navigate to the org.wso2.carbon.esb.connector.googlebooks directory and execute the following maven command:

mvn clean install

This builds the connector and generates a ZIP file named googleBooks-connector-1.0.0.zip in the target directory.

Using the new connector

Now, let's look at how you can use the new connector in a mediation sequence.

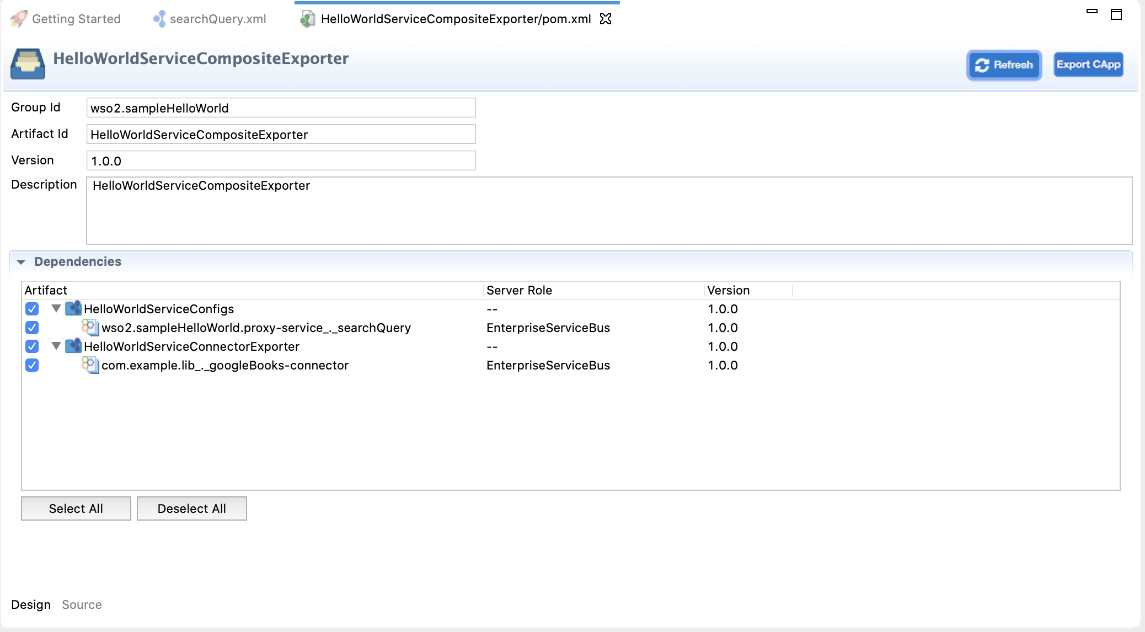
Step 1: Adding the connector to your mediation sequence

1. [Set up WSO2 Integration Studio](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/installing-WSO2-Integration-Studio).
2. [Create an ESB Config project](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/creating-projects/#esb-config-project) and [import the connector](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/creating-artifacts/adding-connectors/#importing-connectors) to your project.

**Tip**

Be sure to select the new googleBooks-connector-1.0.0.zip file from your org.wso2.carbon.esb.connector.googlebooks/target directory.

1. [Create a custom proxy service](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/creating-artifacts/creating-a-proxy-service) named **googlebooks\_listVolume**. In the **Design View**, you will note that the new connector is added to the tool palette.



1. Now, update the proxy service as shown below. You will be defining a mediation logic using the **Propertry** mediator, the new **googleBooks** connector, and the **Respond** mediator:

<?xml version="1.0" encoding="UTF-8"?>

<**proxy** xmlns="http://ws.apache.org/ns/synapse"

name="googlebooks\_listVolume"

transports="https,http"

statistics="disable"

trace="disable"

startOnLoad="true">

<**target**>

<**inSequence**>

<**property** name="searchQuery" expression="json-eval($.searchQuery)"/>

<**googleBooks.listVolume**>

<**searchQuery**>{$ctx:searchQuery}</**searchQuery**>

</**googleBooks.listVolume**>

<**respond**/>

</**inSequence**>

</**target**>

<**description**/>

</**proxy**>

Step 2: Packaging all the artifacts

You need to package the new connector file and the proxy service separately.

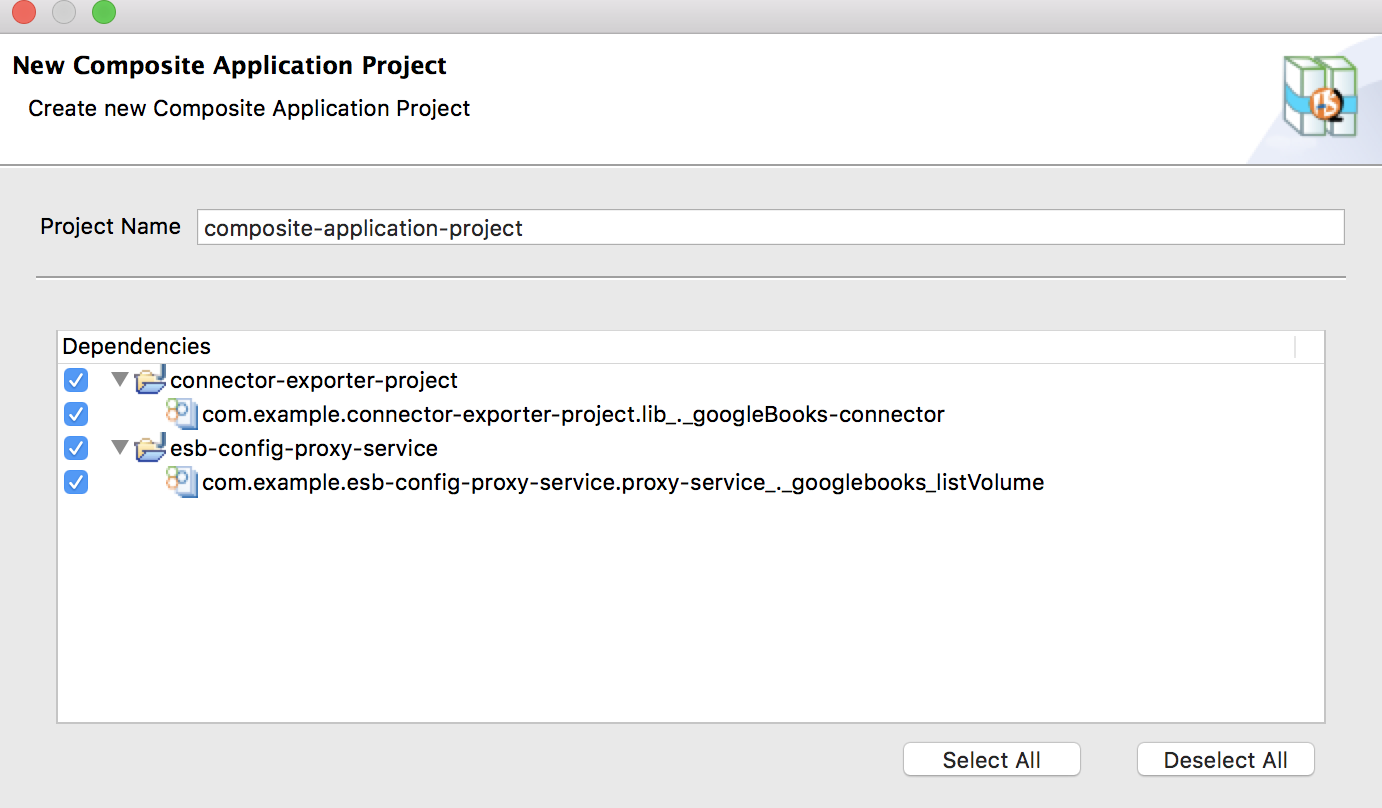
1. Create a **Connector Exporter project** and add the connector.

See the instructions on [packaging a new connector file](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/creating-artifacts/adding-connectors/#packaging-connectors).

1. Create a new **Composite Application project** and add the proxy service as well as the connector as dependencies.

**Tip**

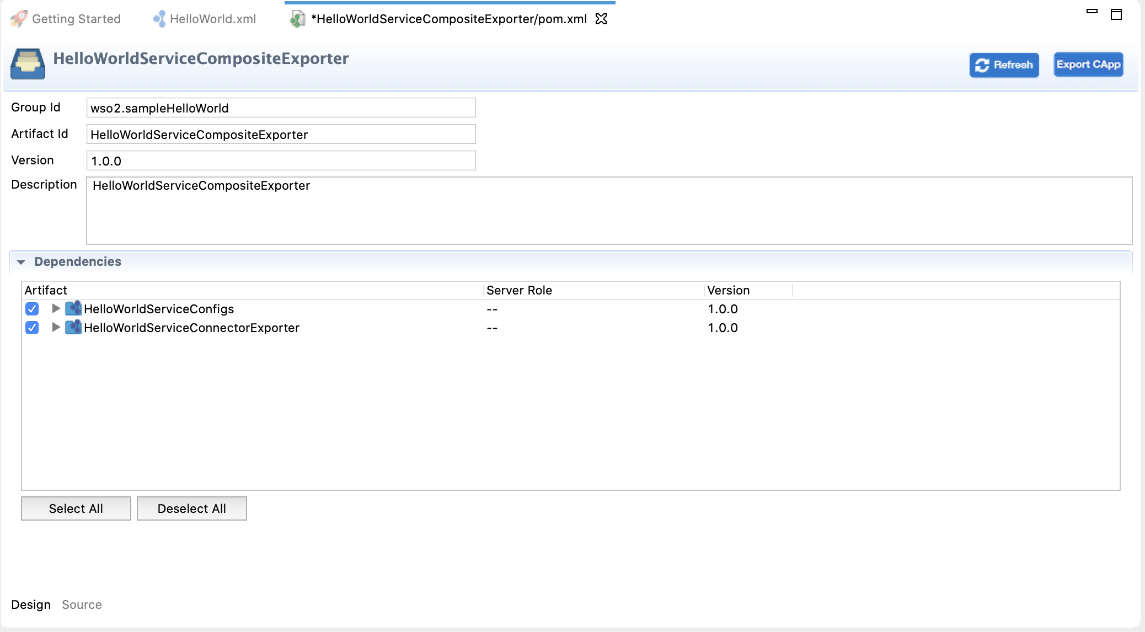
Note that you need to add both the **Connector Exporter project** as well as the **ESB Config project** as dependencies because the connector is referred from the proxy service.



See the instructions on [packaging ESB artifacts](https://ei.docs.wso2.com/en/latest/micro-integrator/develop/packaging-artifacts/#creating-a-new-composite-application).

Step 3: Deploying the artifacts

1. Open the POM file for the composite application project and ensure that the **Connector Exporter** project as well as the **ESB Config** project are selected as dependencies.



1. Right-click the Composite Application project and click **Export Project Artifacts and Run**.

The embedded Micro Integrator will now start and deploy the artifacts.

Step 4: Testing the connector

Post a request to the proxy service using Curl as shown below.

curl -v -X POST -d "{"searchQuery":"rabbit"}" -H "Content-Type: application/json" http://localhost:8290/services/googlebooks\_listVolume

This performs a search and displays a list of volumes that meet the specified search criteria.