**Practical Exercise: Connecting via B2B Protocol**

Training Objective

Learn how to use the File Connector to perform operations in the local file system as well as in a remote server such as FTP and SFTP.

High-level Steps

* Add the File connector into WSO2 Integration Studio.
* Access HTTP API resources.
* Create and read files using the connector.
* Test your configuration.

Detailed Instructions

File Connector Example

File Connector can be used to perform operations in the local file system as well as in a remote server such as FTP and SFTP.

What you'll build

This example describes how to use the File Connector to write messages to local files and then read the files. Similarly, the same example can be configured to communicate with a remote file system (i.e FTP server) easily. The example also uses some other WSO2 mediators to manipulate messages.

An API is exposed to accept XML messages (employee information). When a message is received, it is converted to a CSV message and then stored to a property. A check is done to see if the CSV file (with the same information) exists in the file system. If it does not exist, the connector creates a CSV file with CSV headers included. Then, the connector appends the new CSV entries in the current message to the CSV file. The connector then reads the same CSV file and converts the information back to XML and responds to the client.

Setting up the environment

Create a folder in your local file system with read and write access. This will be your working directory. In this example, it is /Users/hasitha/temp/file-connector-test/dataCollection.

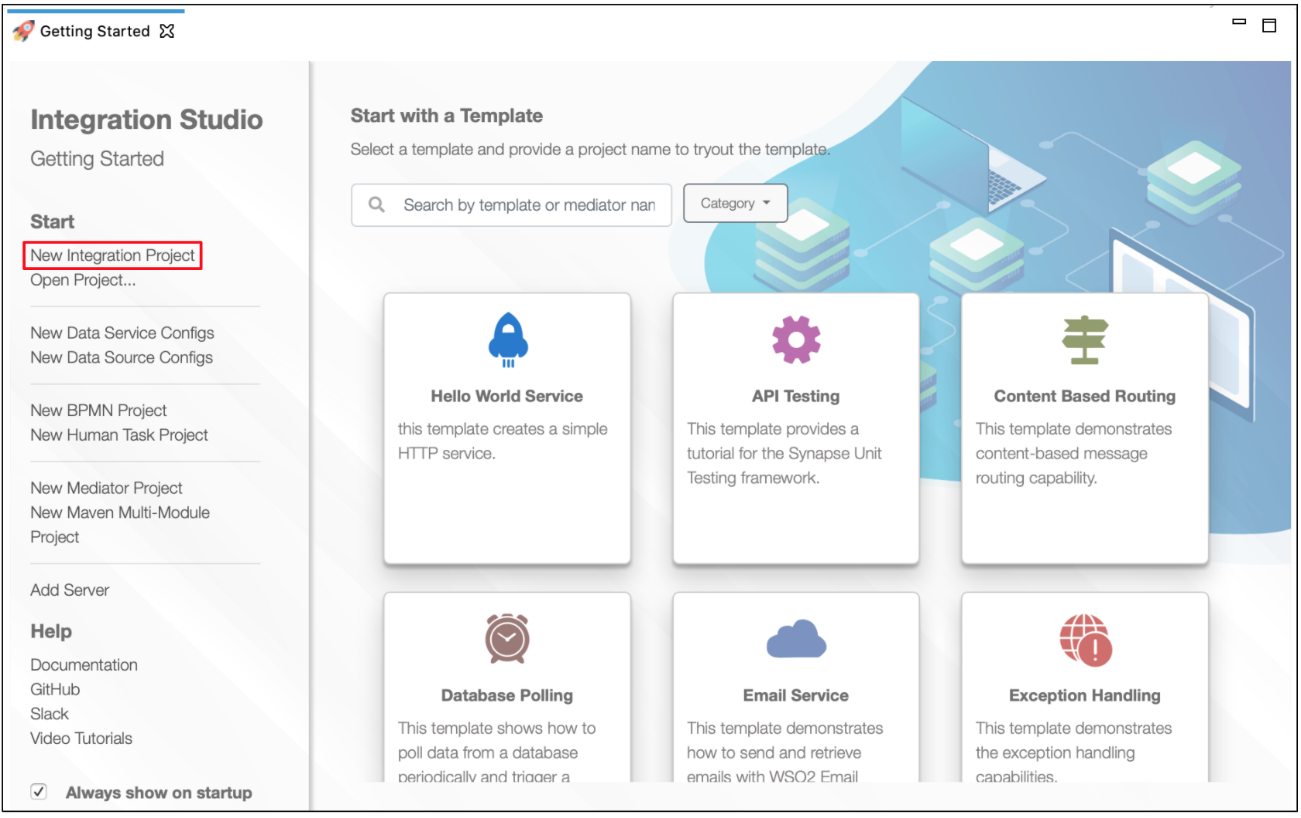
**Note**

If you set up a FTP server, SFTP server, or a Samba server, do the required configurations and select a working directory. Save the host, port, and security related information for future use.

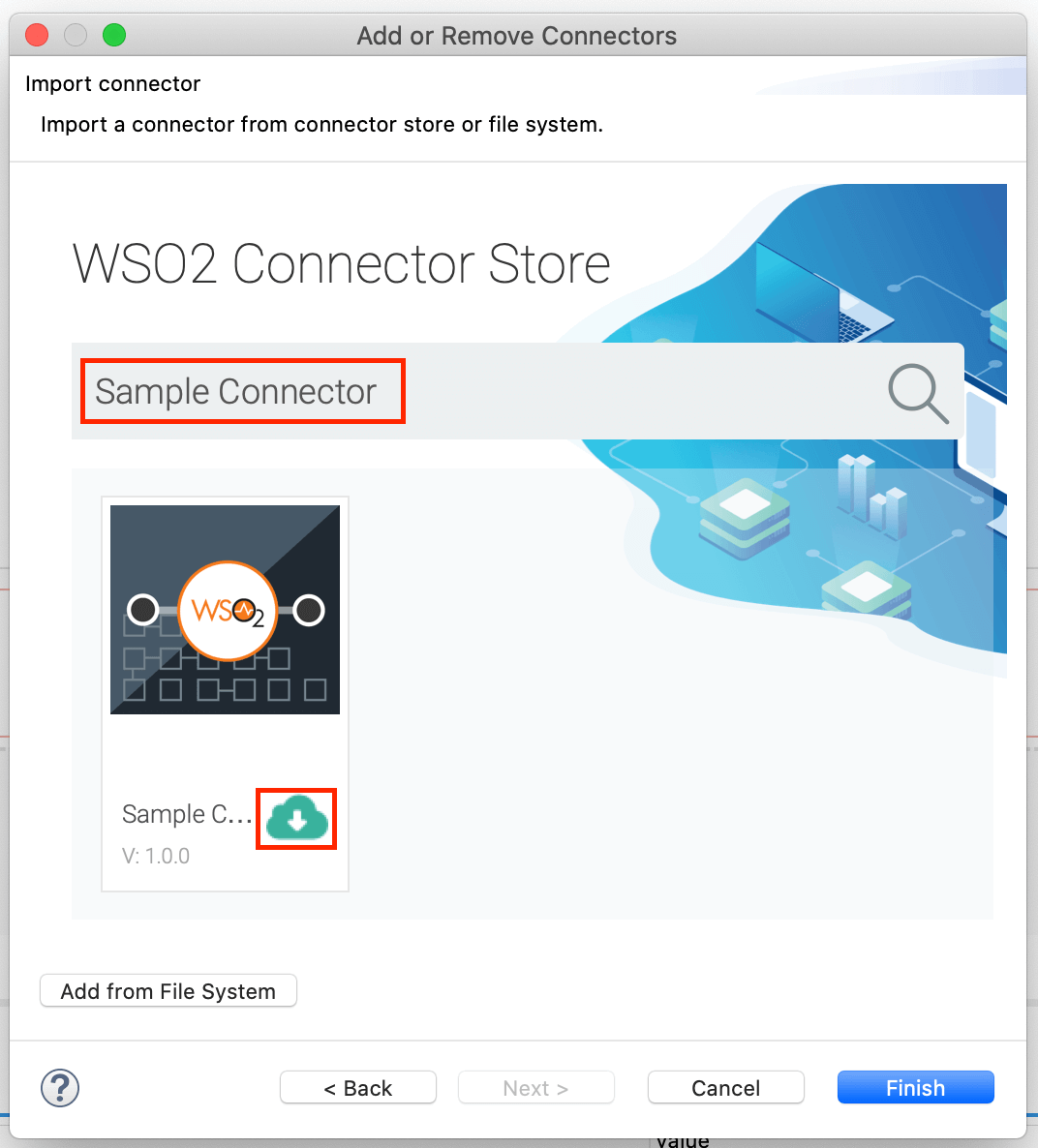
Configure the connector in WSO2 Integration Studio

Follow these steps to set up the Integration Project and the Connector Exporter.

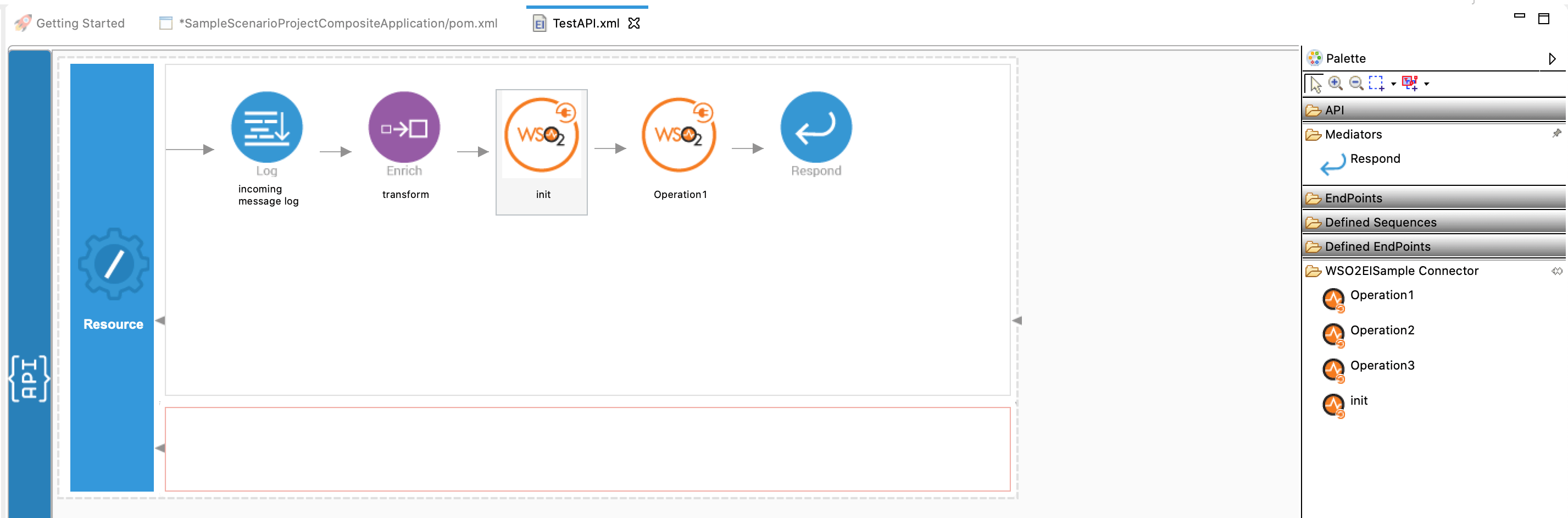
1. Open WSO2 Integration Studio and create an **Integration Project**.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/new-project/new-integration-project.png)

1. Right-click the project that you created and click on **Add or Remove Connector** -> **Add Connector**. You will get directed to the WSO2 Connector Store.
2. Search for the specific connector required for your integration scenario and download it to the workspace.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/search-connector.png)

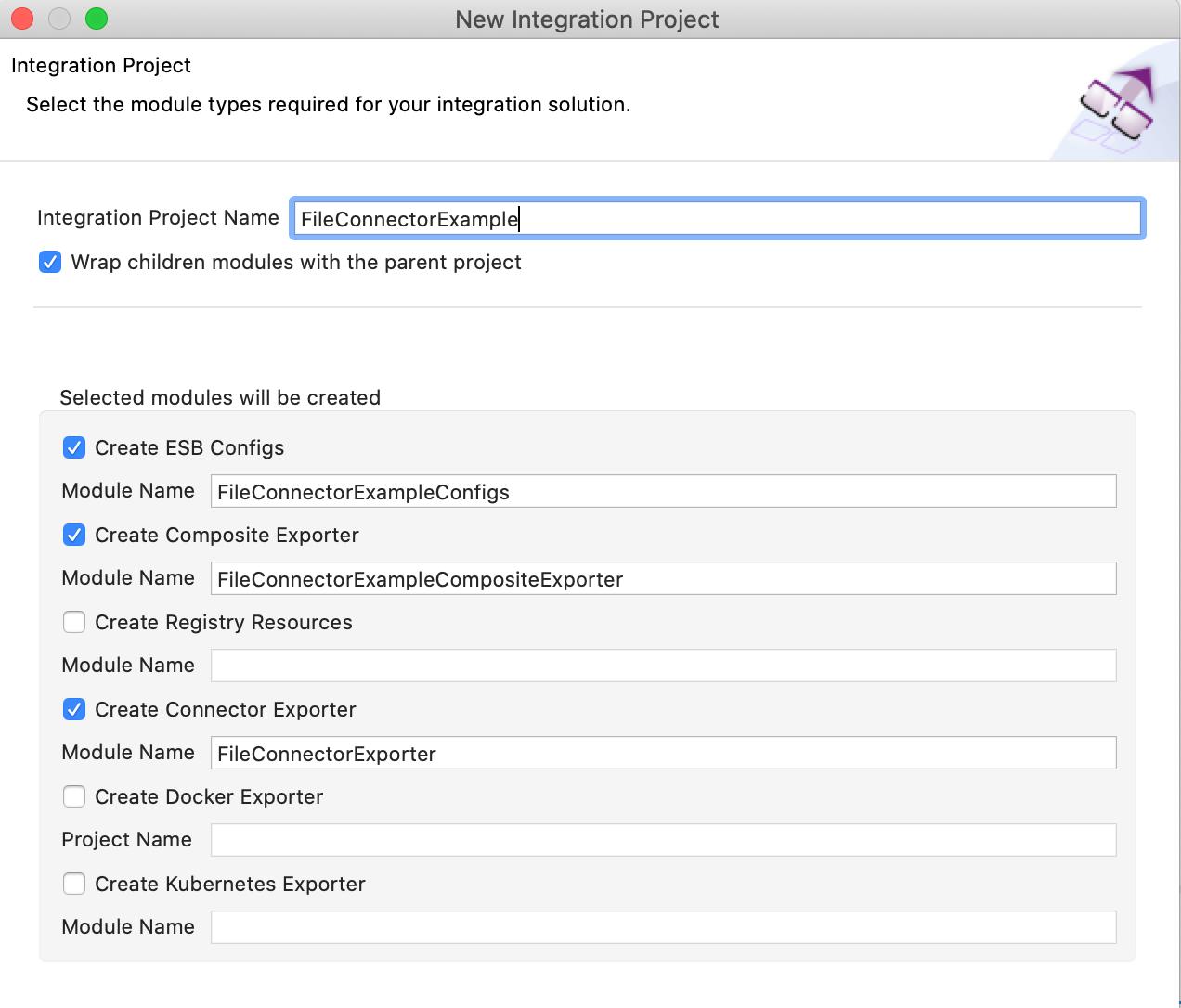
1. Click **Finish**, and your Integration Project is ready. The downloaded connector is displayed on the side palette with its operations.
2. You can drag and drop the operations to the design canvas and build your integration logic.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/drag-connector-operation.png)

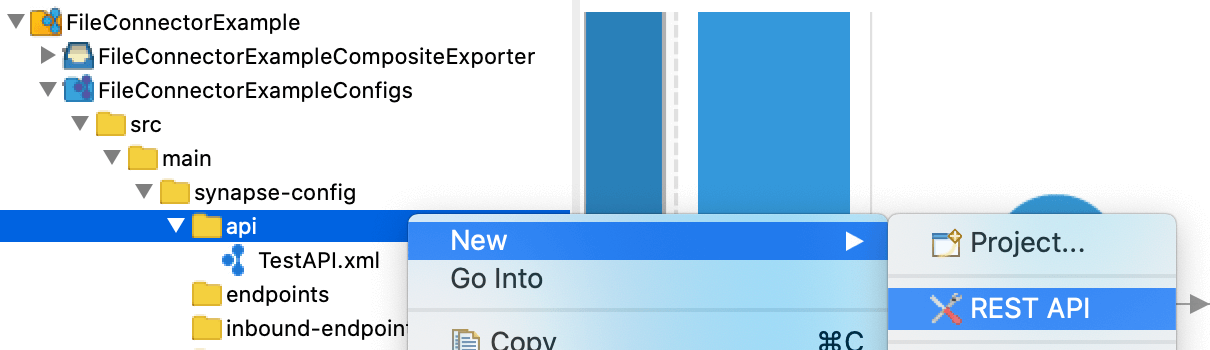
1. Right click on the created Integration Project and select **New** -> **Rest API** to create the REST API.

Creating the Integration Logic

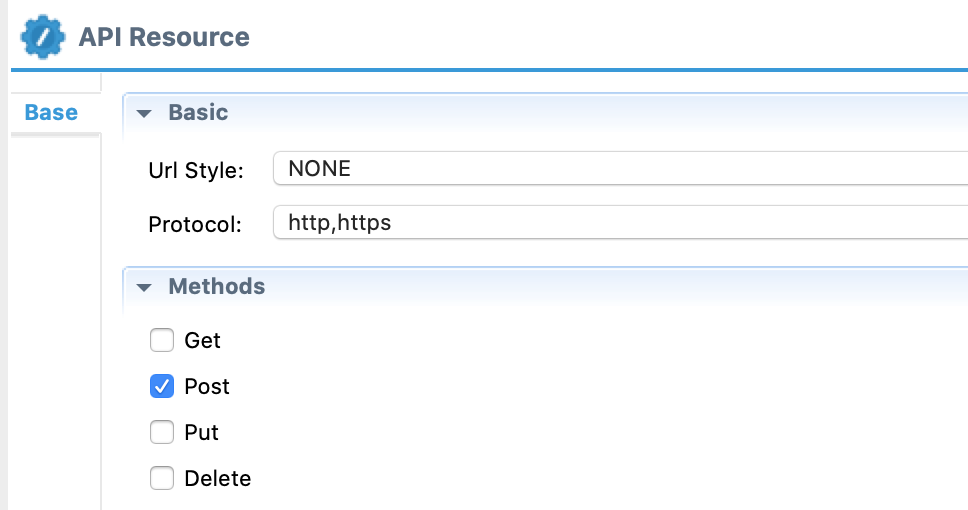
1. Create a new integration project. Be sure to enable a Connector Exporter.



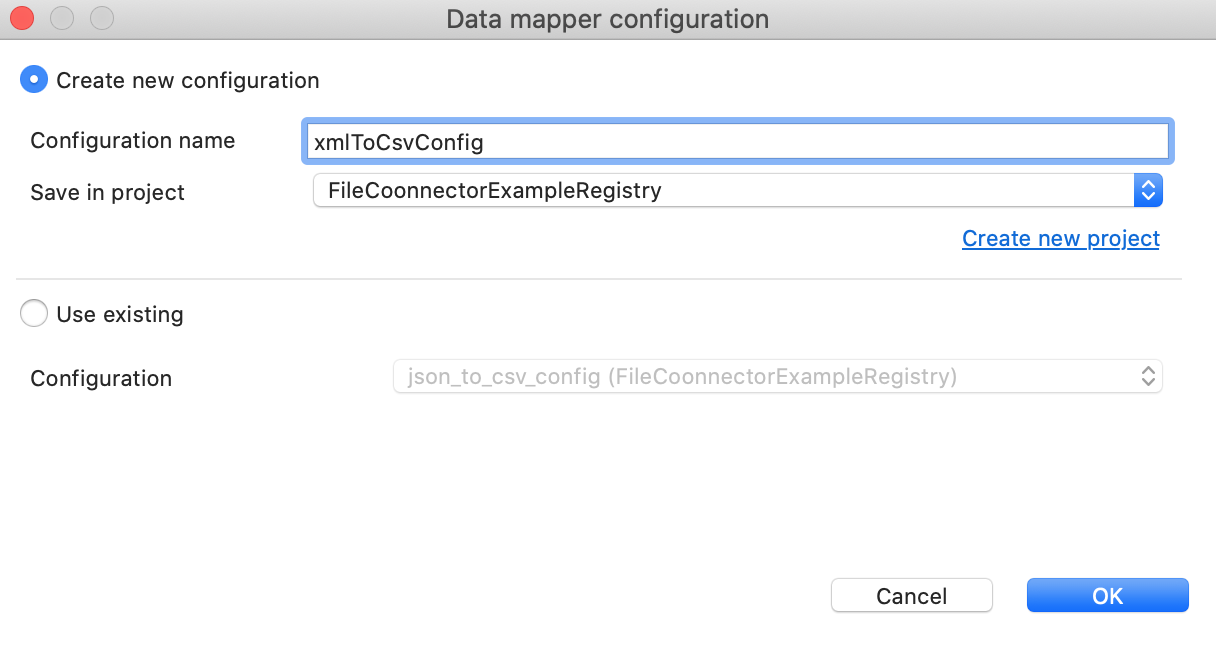
1. Create an API named TestAPI with the /fileTest context. This API will accept employee information.



1. In the default resource of the API, enable POST requests.



1. Add the Log mediator to the design canvas and configure a custom log that indicates when the API receives a message.
2. Add the DataMapper mediator and configure it to transform the incoming XML message to a CSV message.
3. Double-click the Datamapper mediator and add a new transform configuration called 'xmlToCsv'.



1. Save the following content as an XML file. This will be the data input file.

<**test**>

<**information**>

<**people**>

<**person**>

<**name**>Hasitha</**name**>

<**age**>34</**age**>

<**company**>wso2</**company**>

</**person**>

<**person**>

<**name**>Johan</**name**>

<**age**>32</**age**>

<**company**>IBM</**company**>

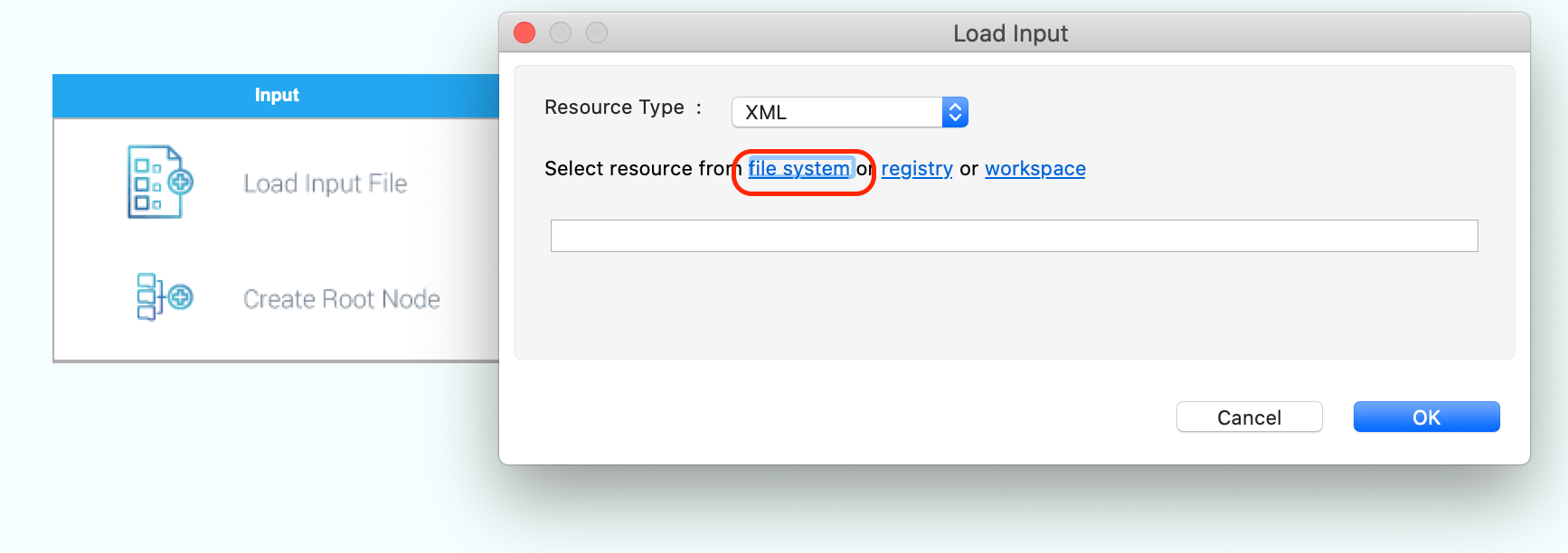
</**person**>

</**people**>

</**information**>

</**test**>

1. Load the input file into the Datamapper config view.



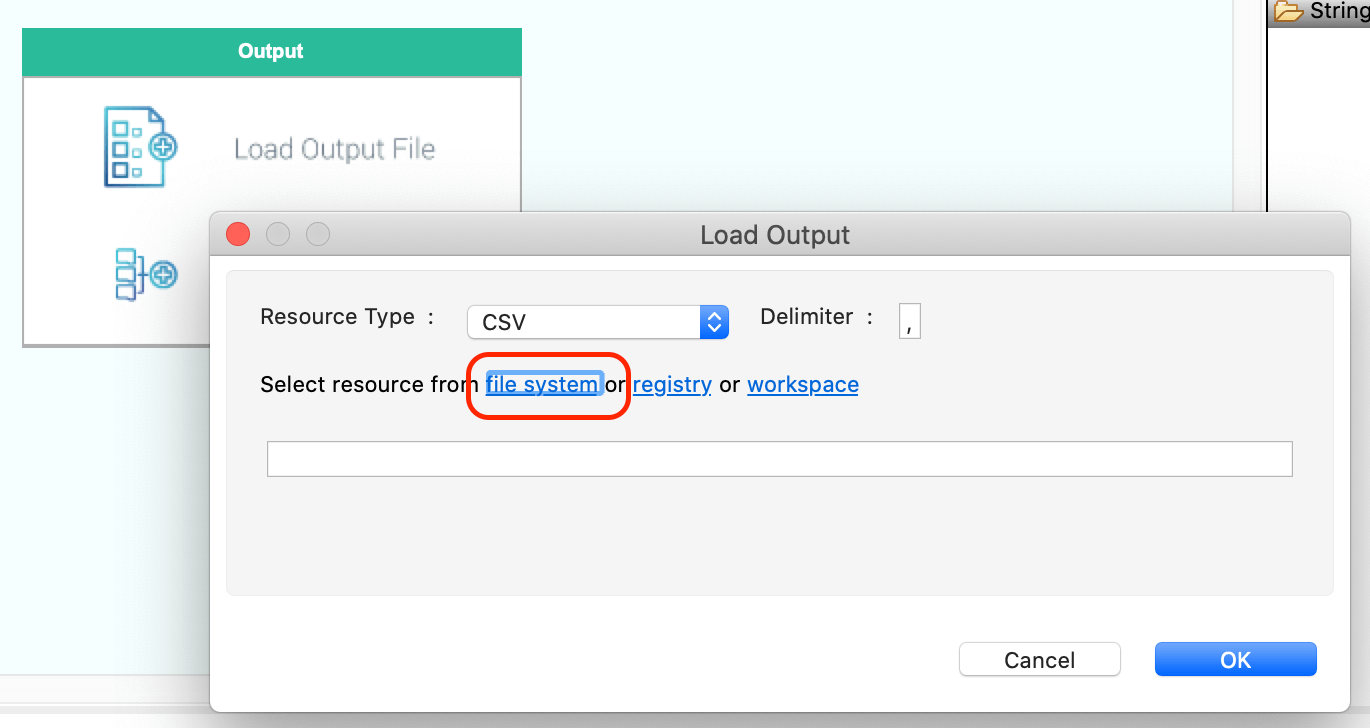
1. Save the following content as a CSV file. This will be the data output file.

Name,Age,Company

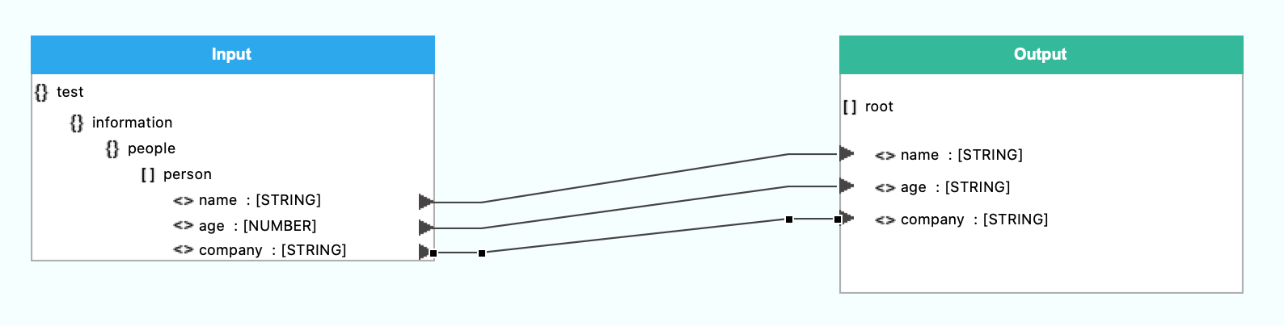
Hasitha,34,wso2

Johan,32,IBM

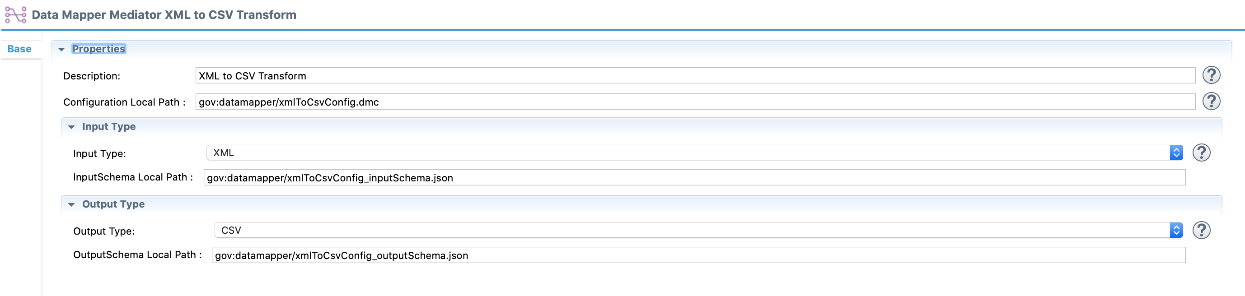
1. Load the output CSV file into the datamapper config view.



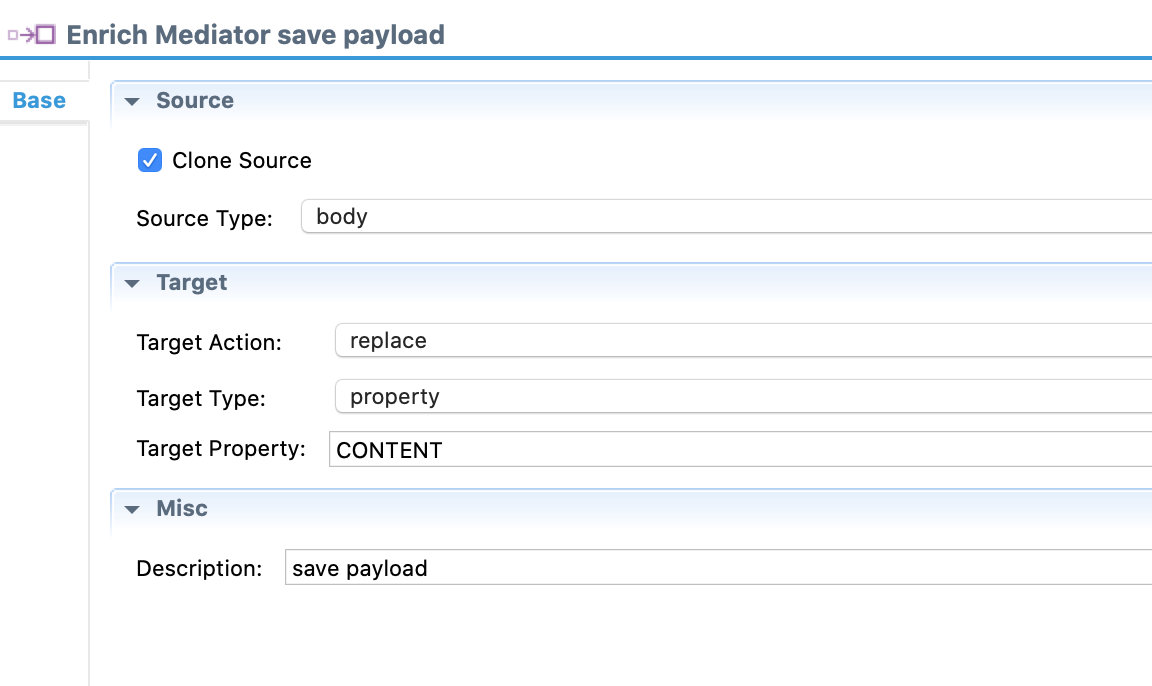
1. Configure the mapping as shown below. Each element in the input should be mapped to the respective element in the output.



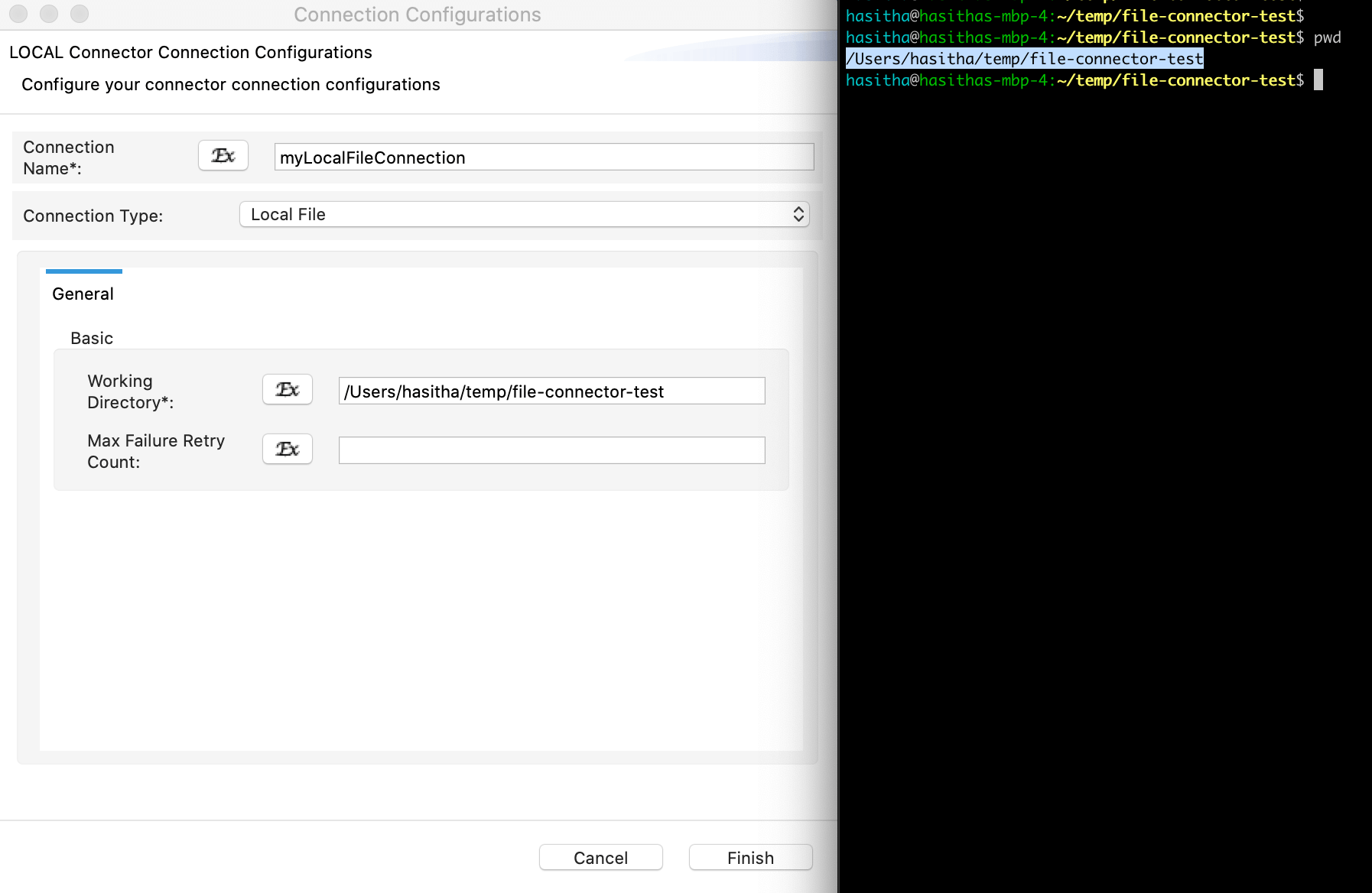
1. Specify the input as XML and output as CSV in the datamapper as shown below.



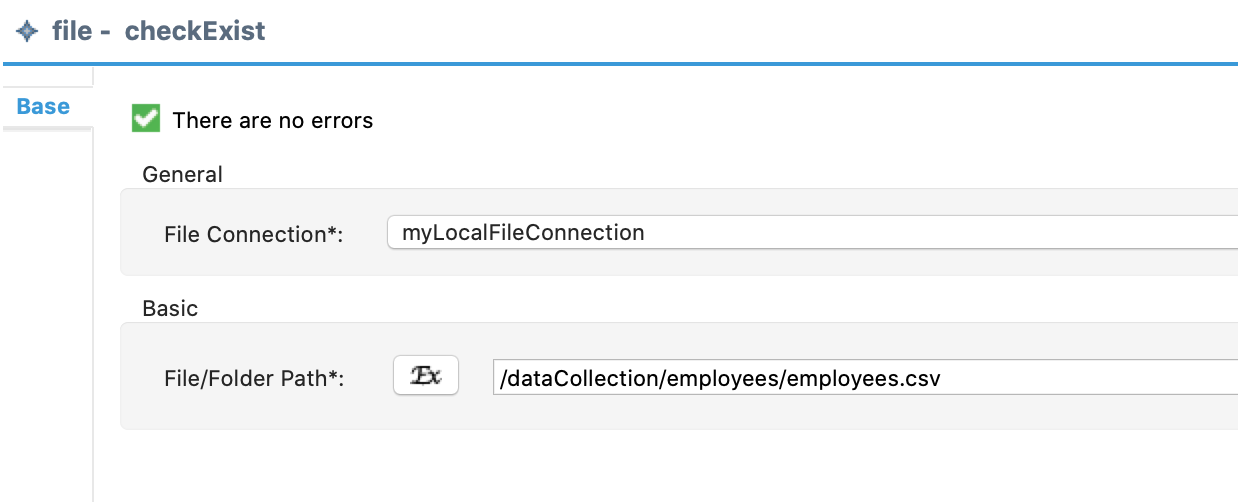
1. Add the Enrich mediator and configure it to save the output generated by the datamapper to a property named CONTENT.



1. Now, let's use the File connector to check if the file containing employee information already exists in the file system.
   1. Add the **checkExist** operation of the File connector to the canvas.
   2. Create a new file connection pointing to the working directory we already set up. Keep this as the File connection for the operation.



* 1. Configure the file path as /dataCollection/employees/employees.csv. This file will store the employee information.

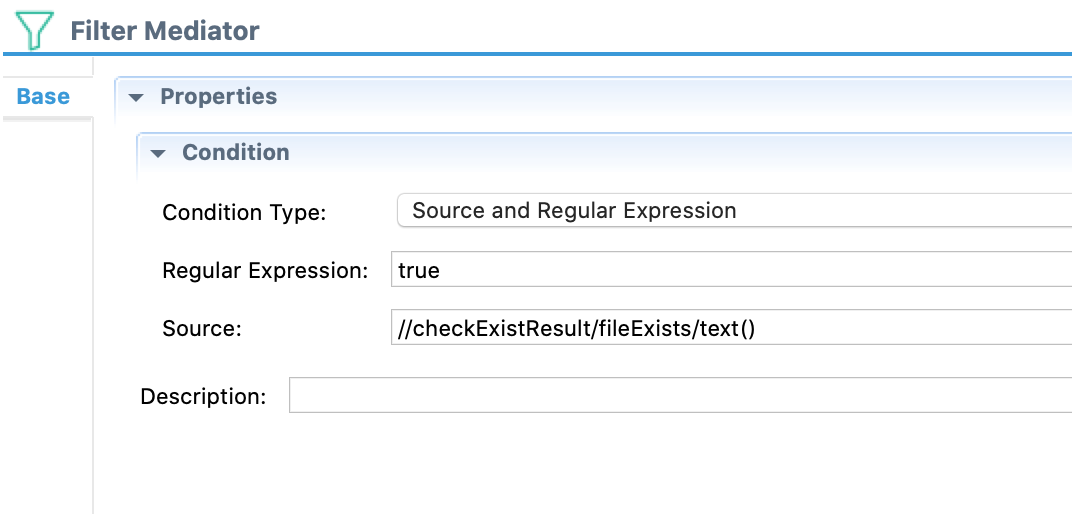


1. Add the Filter mediator to branch out the logic based on the result from the File connector’s checkExist operation.

**Note**

If the file does not exist, the File connector’s **write** operation (which we configure later) will create the file.

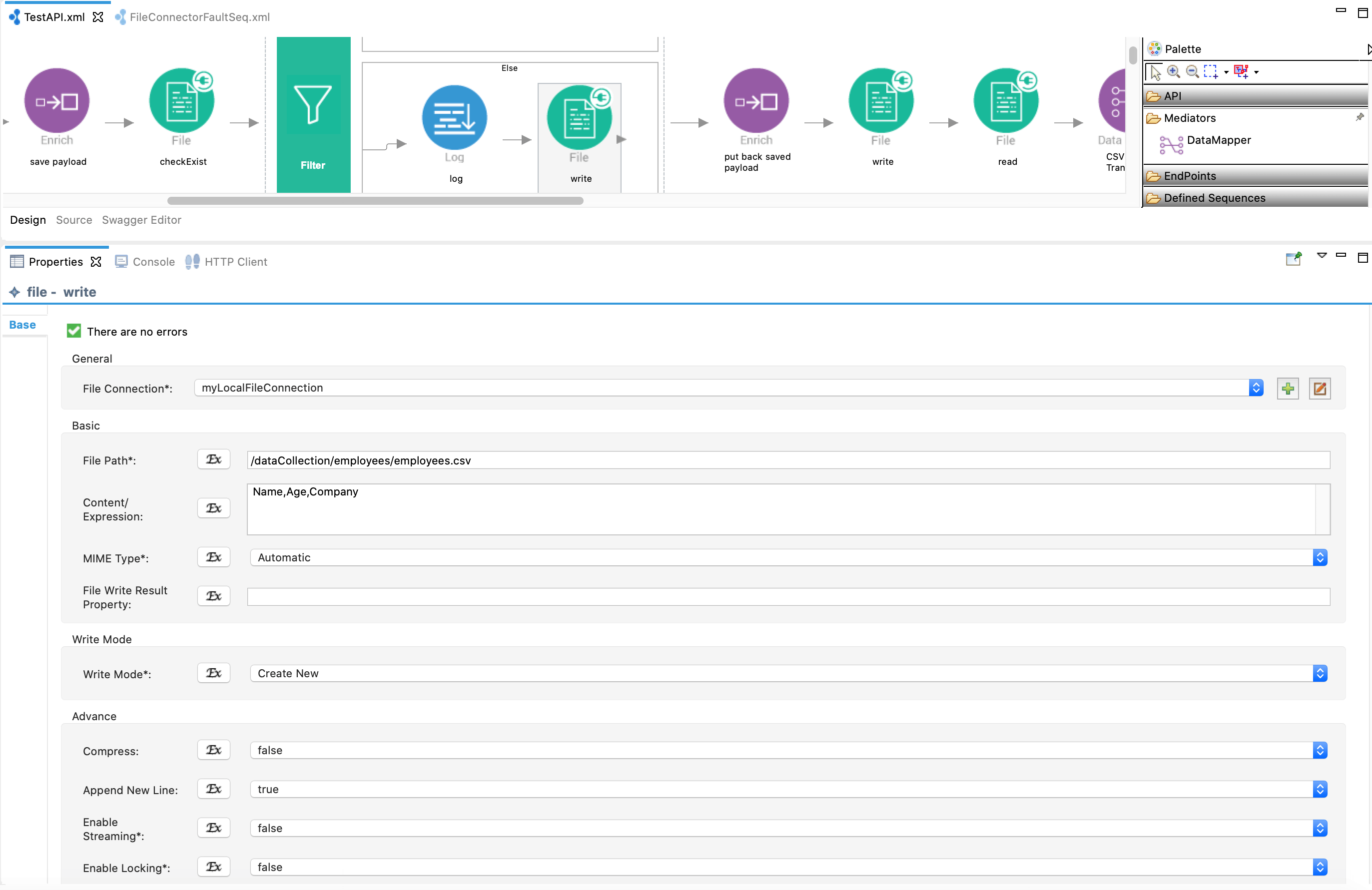
* 1. Click the Filter mediator and define the filter condition as shown below.



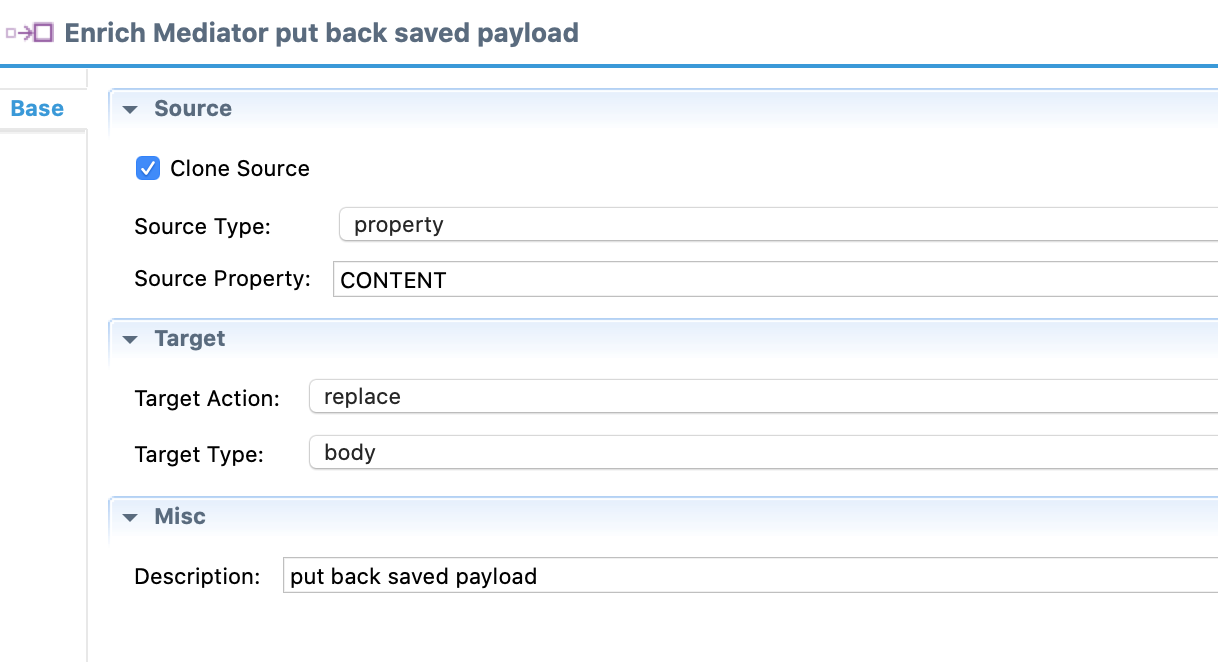
* 1. Inside the “else” block, add the File Connector's **write** operation and configure it to write the static content of CSV file headers: “Name,Age,Company”.

**Note**

Be sure to append a new line at the end of the file. The **Write Mode** needs to be Create New.



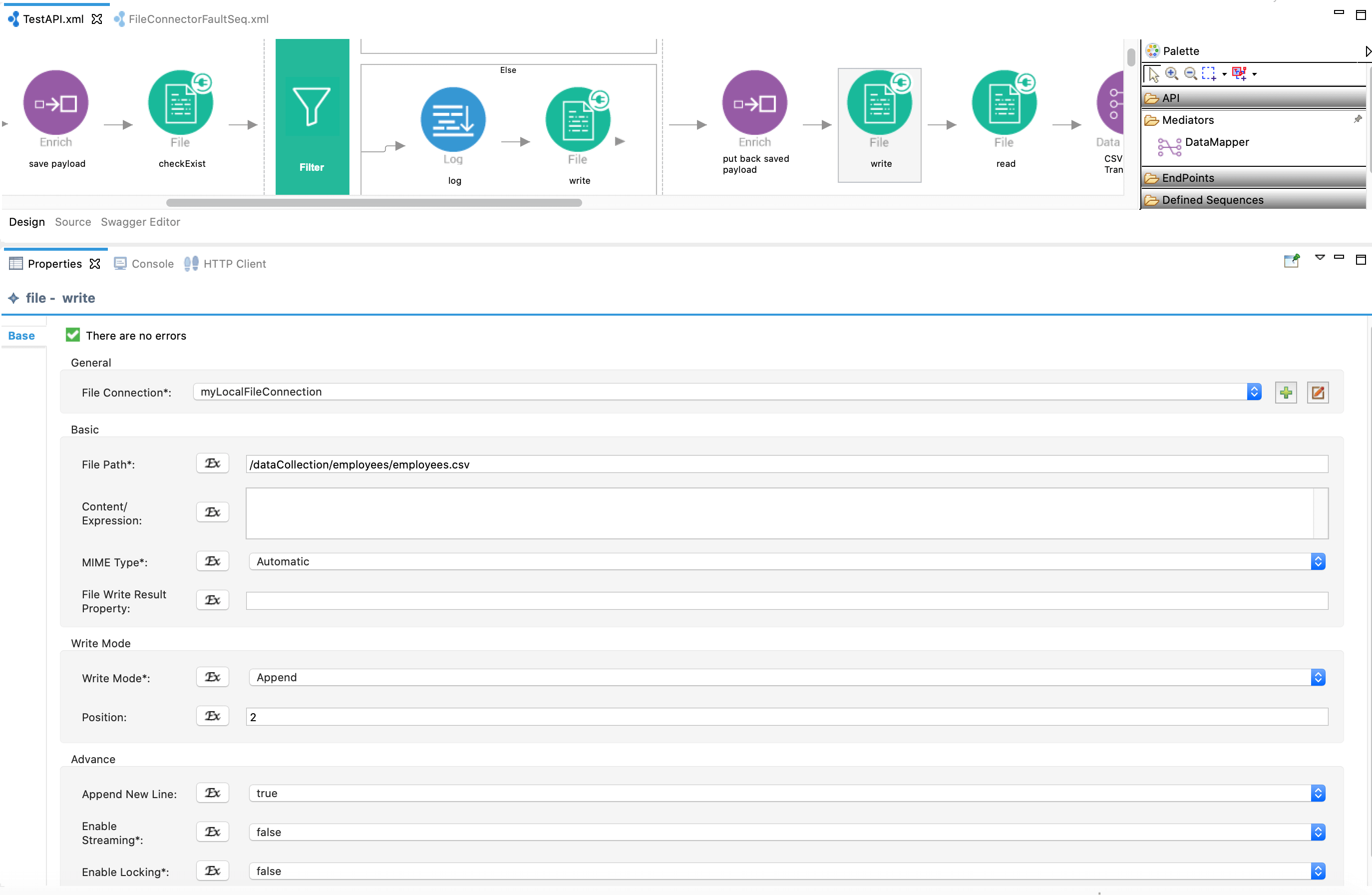
1. After the Filter mediator, use the Enrich mediator again to put back the saved payload into the message payload.



1. Add the File connector’s **write** operation again and configure it to append the CSV message to the existing file. The **Write Mode** needs to be 'Append'.

**Note**

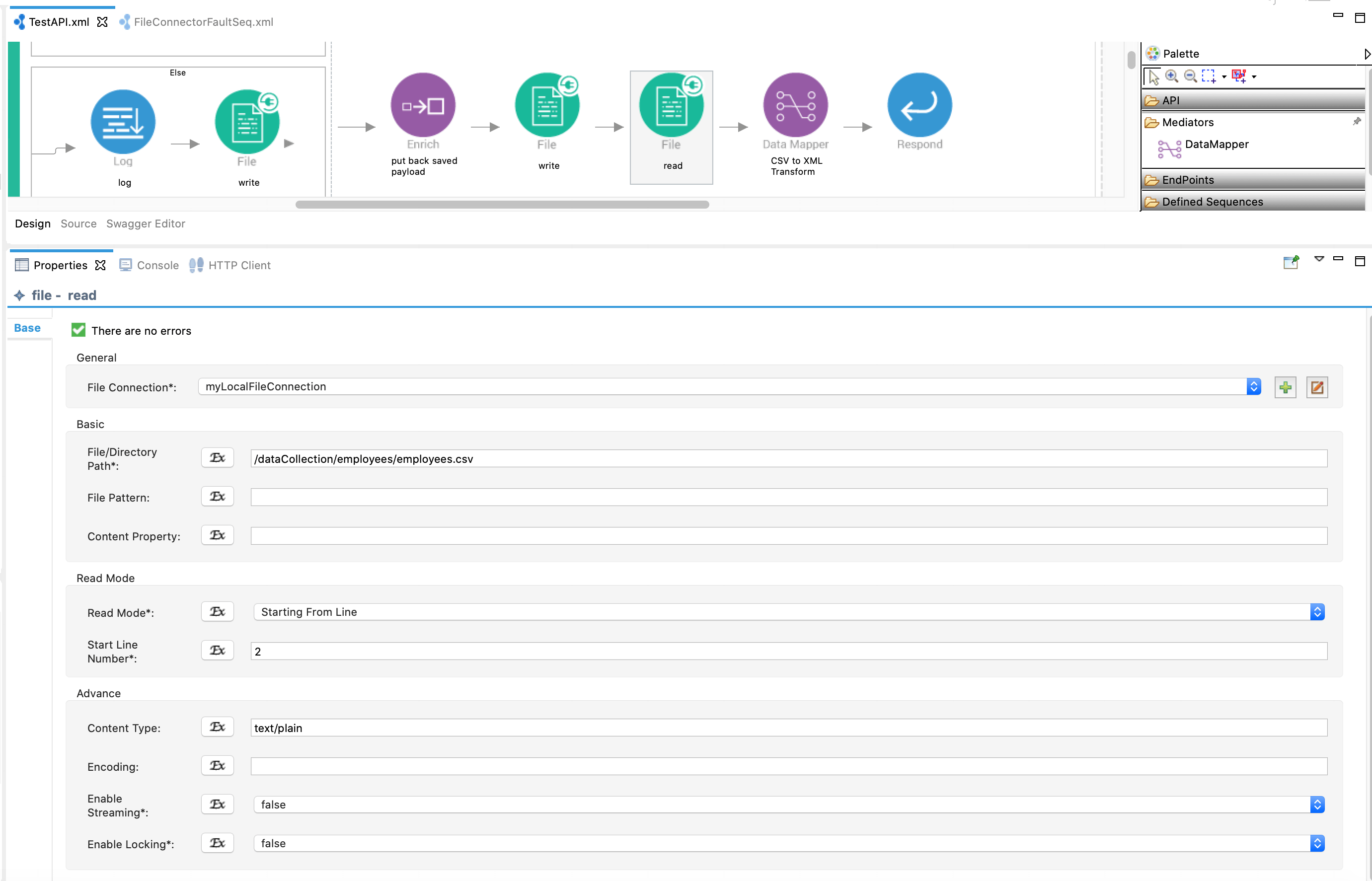
As we need the newest message on the top, always append to line number 2.



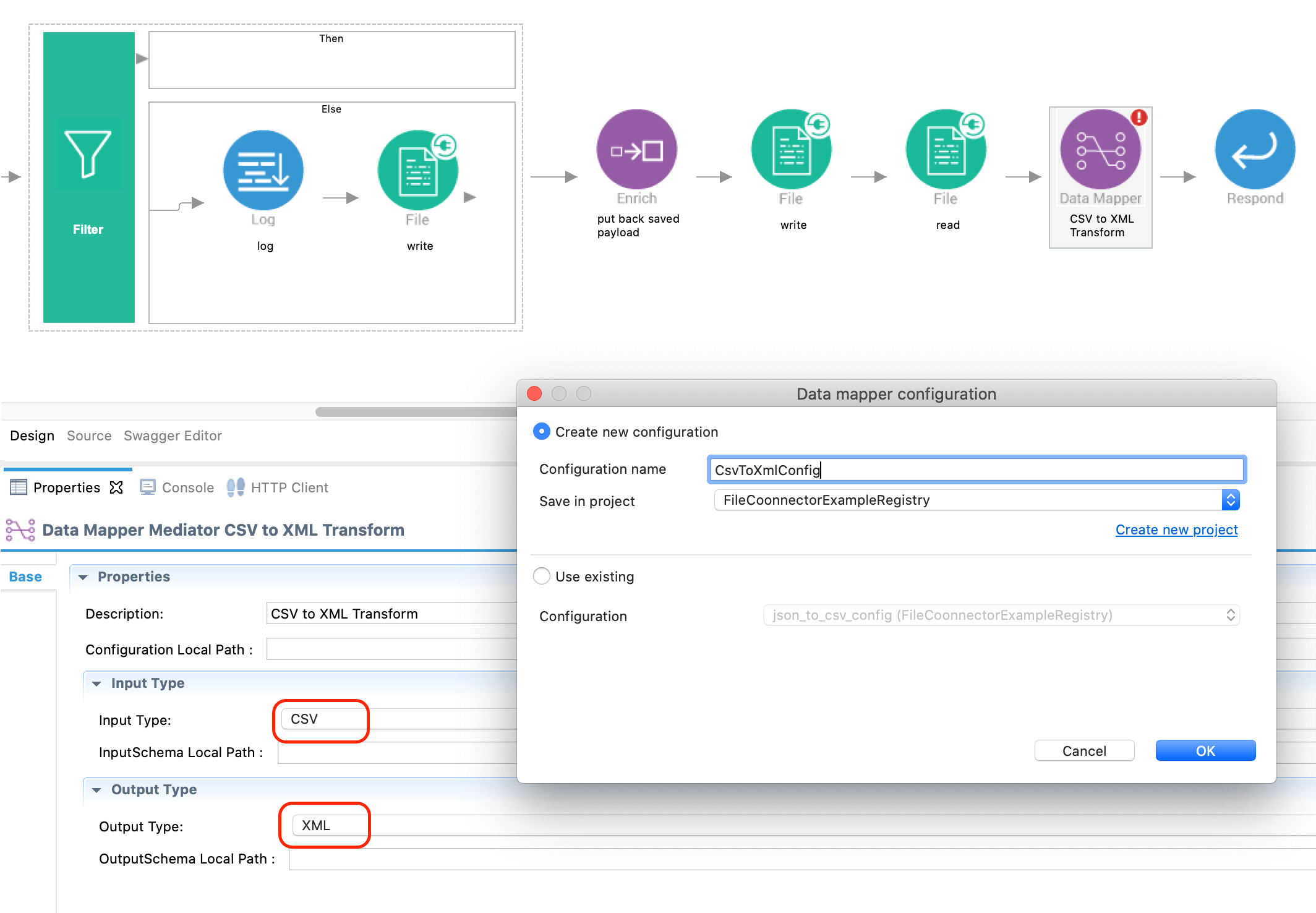
1. Add the File connector’s **read** operation and configure it to read the same CSV file.

**Note**

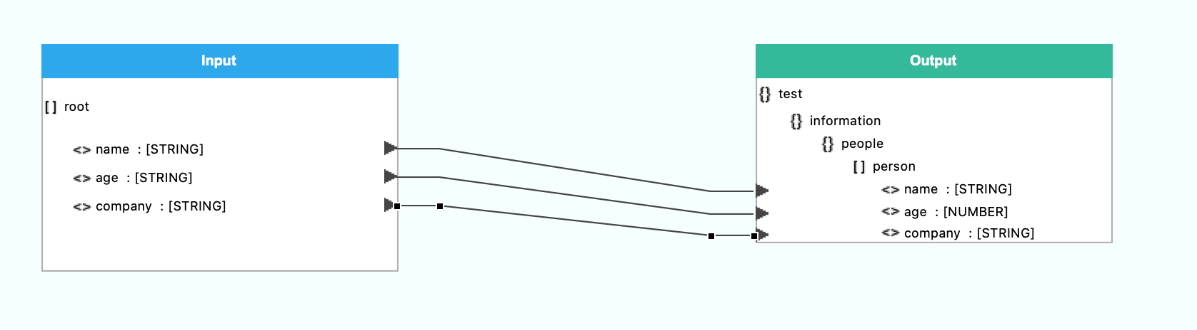
The file reading will start from line number 2. The content is read as a text message.



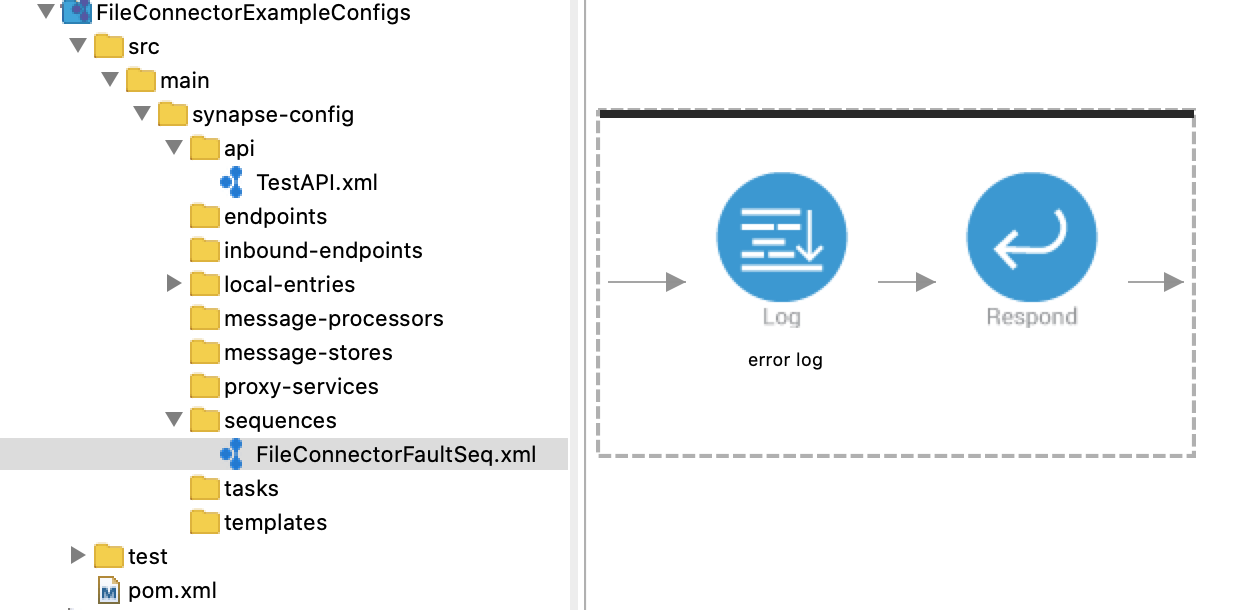
1. Add the Datamapper mediator again and configure it to convert the CSV message (after reading) back to XML.
   1. Double-click the data mapper and add a new configuration called 'csvToXml'.



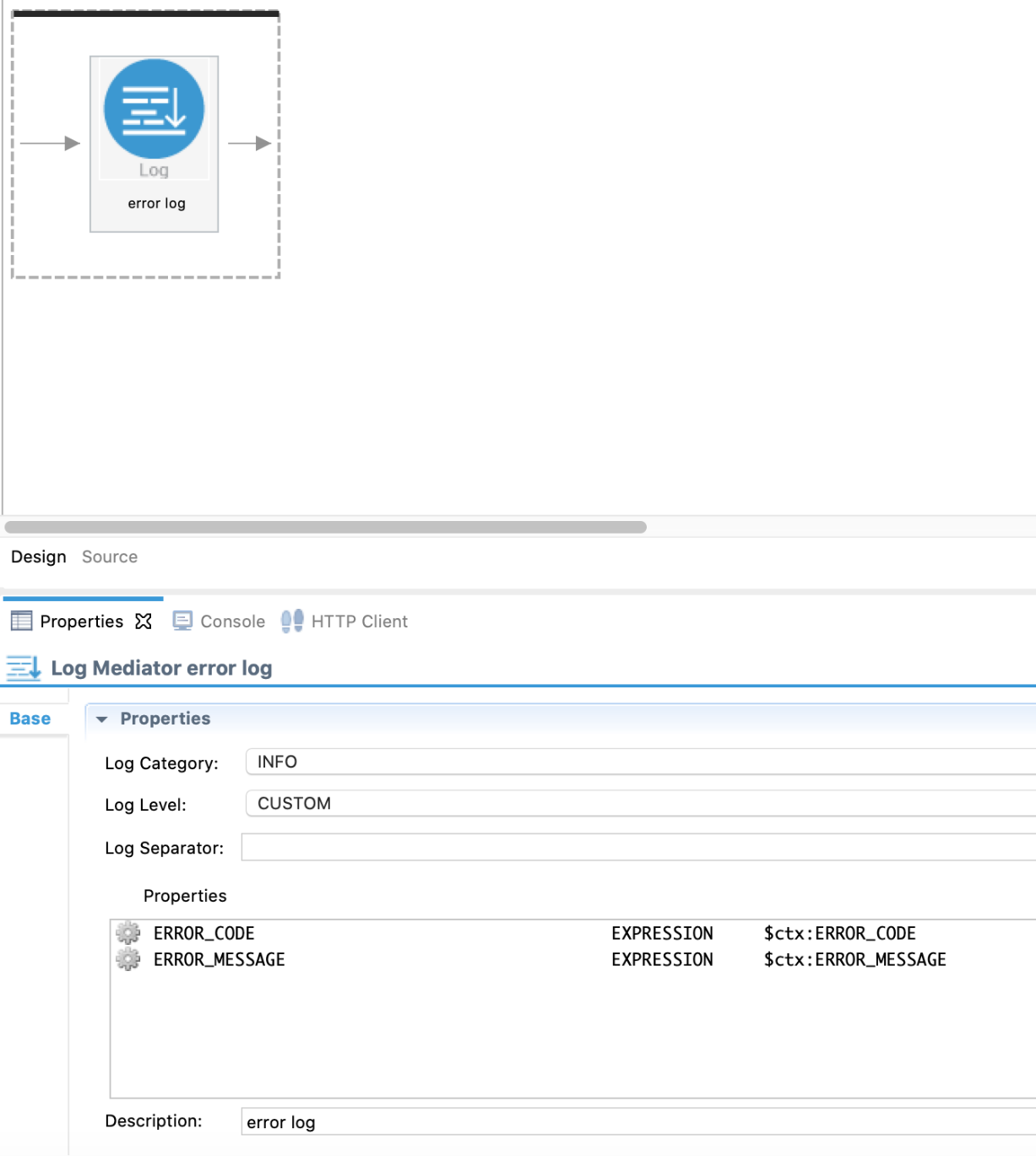
* 1. This time, the mapping should be from CSV to XML.



1. Finally, use the **Respond** mediator to send the transformed message to the API caller.
2. Now, let's configure a fault sequence to generate an error message when an error occurs in the message flow.
   1. Create a fault sequence with a **Log** mediator and **Respond** mediator.



* 1. Configure the Log mediator generate a custom error.



* 1. Add the fault sequence to the API resource as its fault sequence.

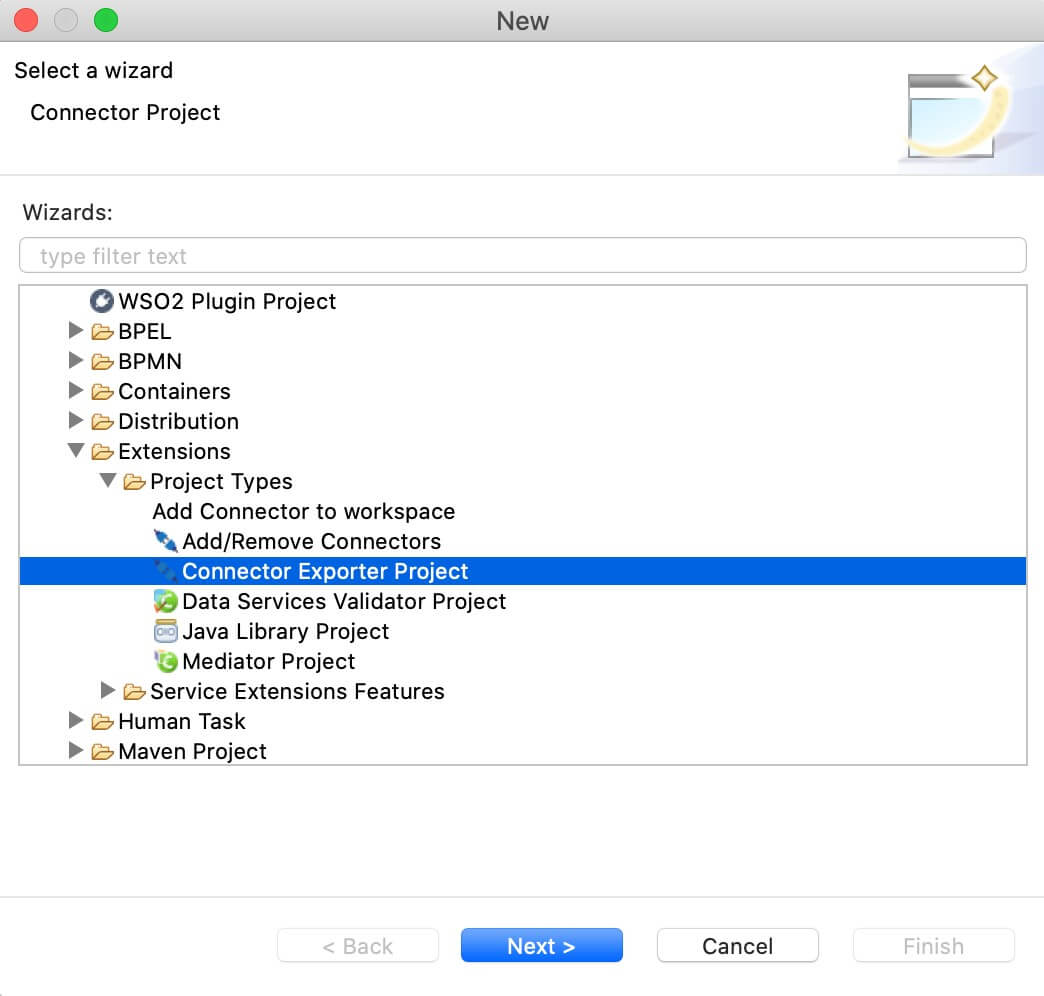
Exporting Integration Logic as a CApp

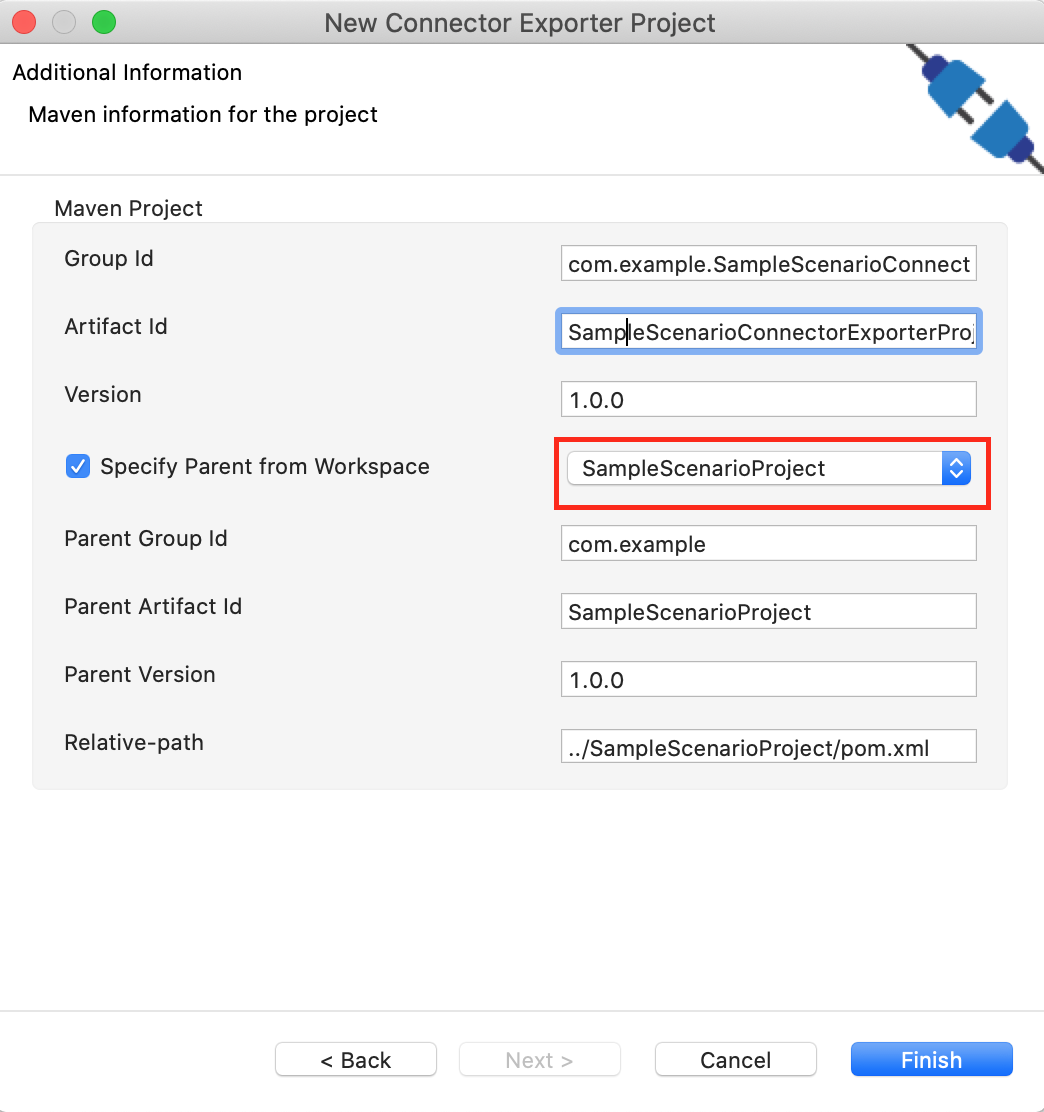
**CApp (Carbon Application)** is the deployable artifact on the integration runtime. Let us see how we can export integration logic we developed into a CApp along with the connector.

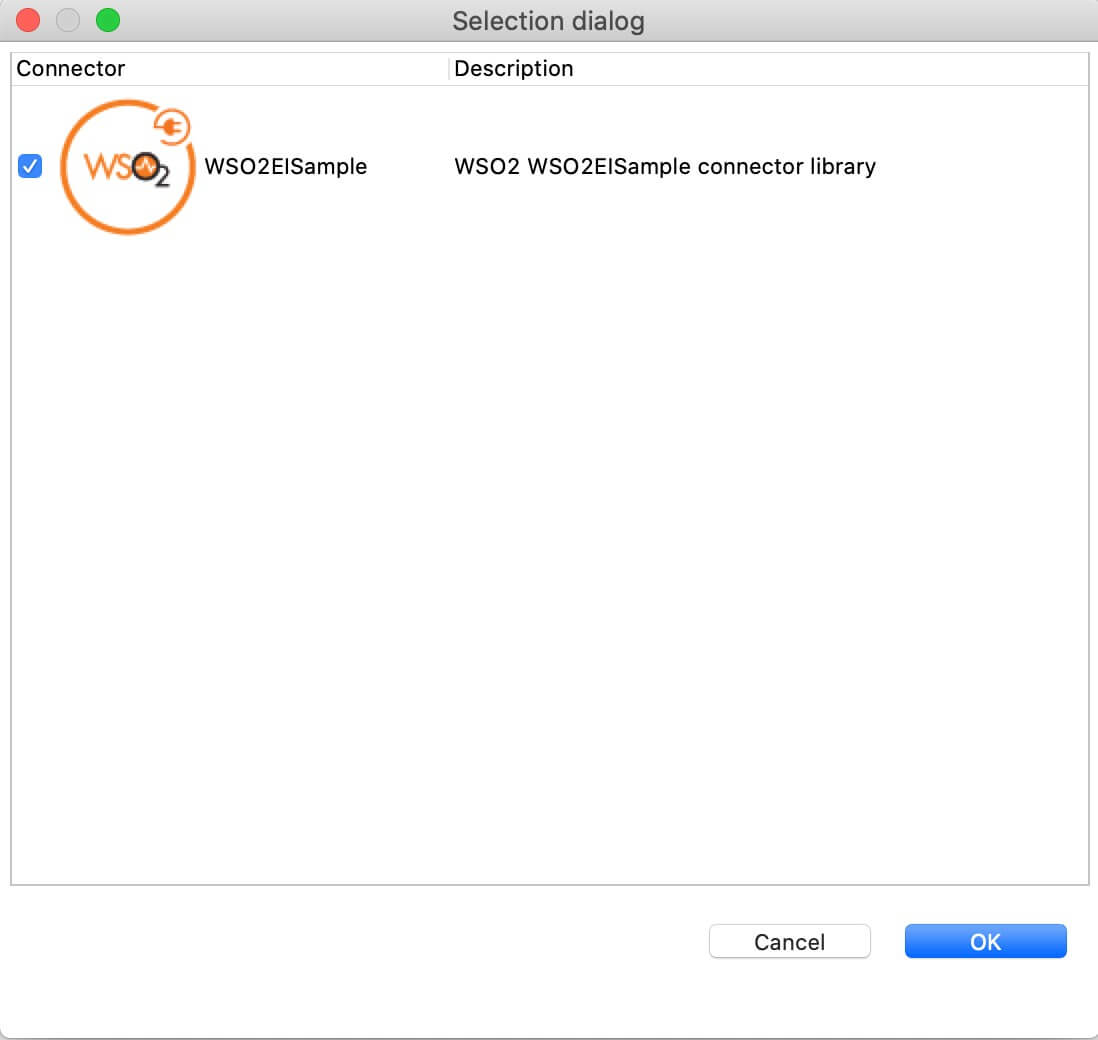
Creating Connector Exporter Project

To bundle a Connector into a CApp, a Connector Exporter Project is required.

1. Navigate to **File** -> **New** -> **Other** -> **WSO2** -> **Extensions** -> **Project Types** -> **Connector Exporter Project**.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/connector-exporter-project-1.jpg)

1. Enter a name for the Connector Exporter Project.
2. In the next screen select, **Specify the parent from workspace** and select the specific Integration Project you created from the dropdown.  
   [](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/connector-exporter-project-naming.png)
3. Now you need to add the Connector to Connector Exporter Project that you just created. Right-click the Connector Exporter Project and select, **New** -> **Add Remove Connectors** -> **Add Connector** -> **Add from Workspace** -> **Connector**
4. Once you are directed to the workspace, it displays all the connectors that exist in the workspace. You can select the relevant connector and click **Ok**.

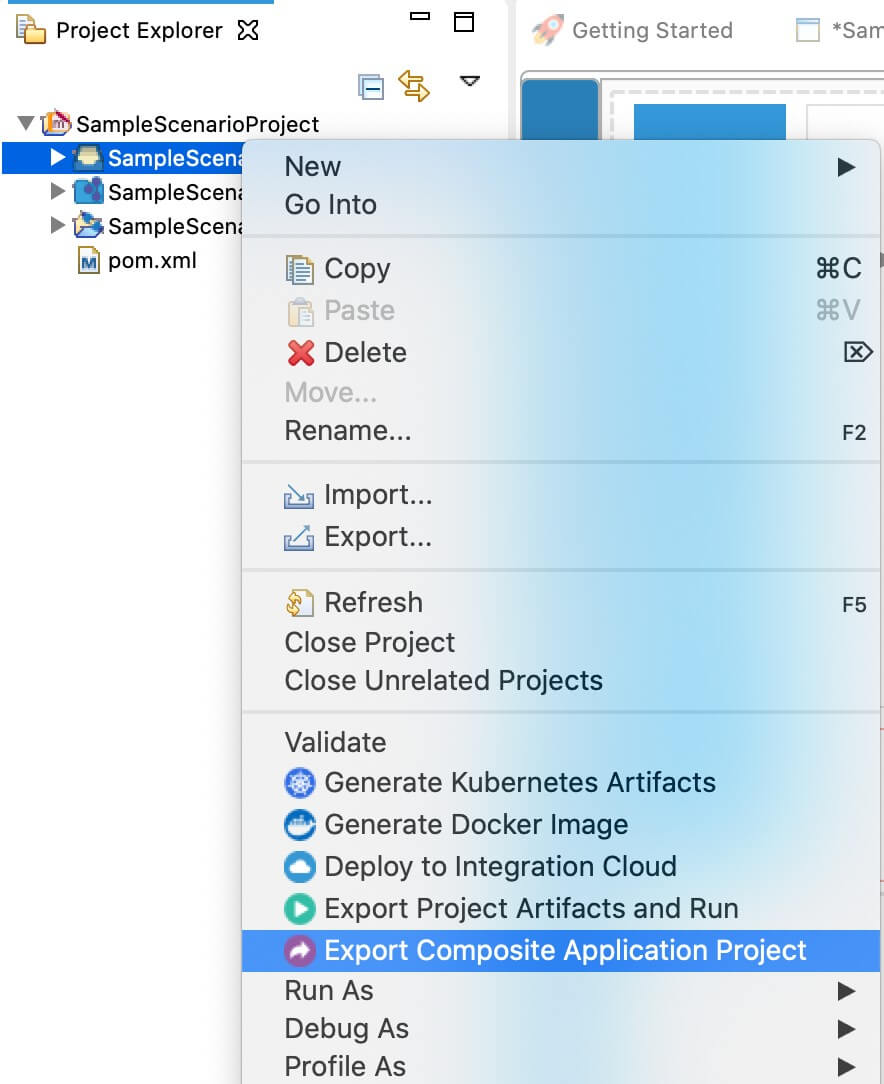
[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/adding-connector-to-exporter-project-3.jpg)

[Creating a Composite Application Project](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/adding-connector-to-exporter-project-3.jpg)

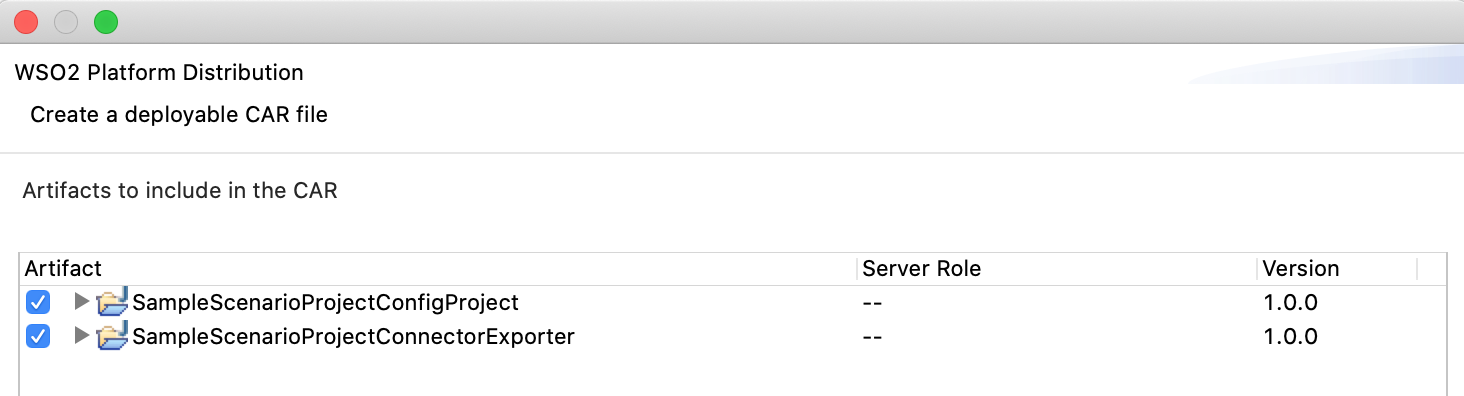
To export the Integration Project as a CApp, a Composite Application Project needs to be created. Usually, when an Integration project is created, this project can be created as part of that project by Integration Studio. If not, you can specifically create it by navigating to **File** -> **New** -> **Other** -> **WSO2** -> **Distribution** -> **Composite Application Project**.

Exporting the Composite Application Project

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



1. Select an **Export Destination** where you want to save the .car file.
2. In the next **Create a deployable CAR file** screen, select both the created Integration Project and the Connector Exporter Project to save and click **Finish**. The CApp is created at the specified location provided at the previous step.

[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/saving-projects.png)

Deployment

Follow these steps to deploy the exported CApp in the integration runtime.

**Deploying on Micro Integrator**

You can copy the composite application to the <PRODUCT-HOME>/repository/deployment/server/carbonapps folder and start the server. Micro Integrator will be started and the composite application will be deployed.

You can further refer the application deployed through the CLI tool. See the instructions on [managing integrations from the CLI](https://apim.docs.wso2.com/en/latest/install-and-setup/setup/api-controller/managing-integrations/managing-integrations-with-ctl/).

Testing

1. Create a file called data.json with the following payload.

**Note**

When you configuring this source parameter in the Windows operating system, set this property as <source>C:\\Users\Name\Desktop\Salesforcebulk-connector\create.txt</source>.

<**test**>

<**information**>

<**people**>

<**person**>

<**name**>Hasitha</**name**>

<**age**>34</**age**>

<**company**>wso2</**company**>

</**person**>

<**person**>

<**name**>Johan</**name**>

<**age**>32</**age**>

<**company**>IBM</**company**>

</**person**>

<**person**>

<**name**>Bob</**name**>

<**age**>30</**age**>

<**company**>Oracle</**company**>

</**person**>

<**person**>

<**name**>Alice</**name**>

<**age**>28</**age**>

<**company**>Microsoft</**company**>

</**person**>

<**person**>

<**name**>Anne</**name**>

<**age**>30</**age**>

<**company**>Google</**company**>

</**person**>

</**people**>

</**information**>

</**test**>

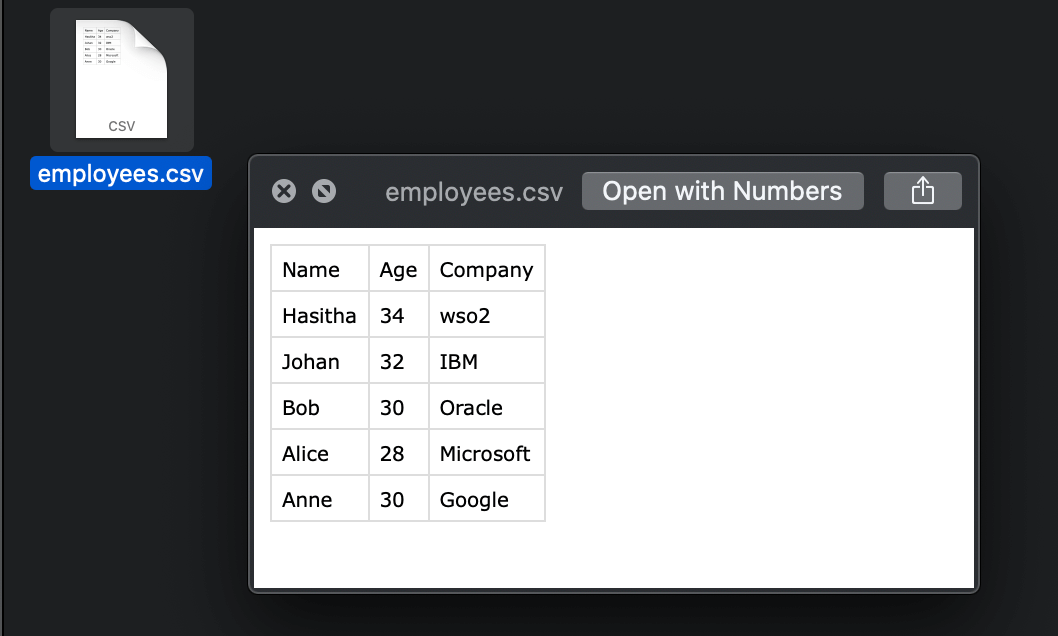
1. Invoke the API as shown below using the curl command.

**Info**

Curl Application can be downloaded from [here](https://curl.haxx.se/download.html).

curl -H "Content-Type: application/xml" --request POST --data @body.json http://10.100.5.136:8290/fileconnector/create

1. Check the file system to verify that the CSV file has been created.



1. If you invoke the API again with a different set of employees, the new employees will get appended to the same file. The response you receive will include all the employees that were added from both messages.

In this example, the File connector was used to create a file, write to a file, and to read a file. By blending these capabilities together with other powerful message manipulation features of WSO2, it is possible to define a working scenario in minutes. The File connector has many more functionalities. Refer the [File Connector reference guide](https://apim.docs.wso2.com/en/4.1.0/reference/connectors/file-connector/file-connector-config/) for more information.

What's Next

* You can deploy and run your project on Docker or Kubernetes. See the instructions in [Running the Micro Integrator on Containers](https://apim.docs.wso2.com/en/4.1.0/install-and-setup/installation/run_in_containers).
* To customize this example for your own scenario, see [File Connector Configuration](https://apim.docs.wso2.com/en/4.1.0/reference/connectors/file-connector/file-connector-config/) documentation for all operation details of the connector.