**Practical Exercise: Connecting Web APIs/Cloud Services**

Training Objective

Learn how to connect the Micro Integrator with a third-party application: Use the Email

connector as an example. Perform operations using the following protocols: SMTP, IMAP, and POP3.

High-level Steps

* Add the Email connector to WSO2 Integration Studio.
* Connect to EmailConnector API.
* Send and retrieve emails using the connector.
* Test your configuration.

Detailed Instructions

Email Connector Example

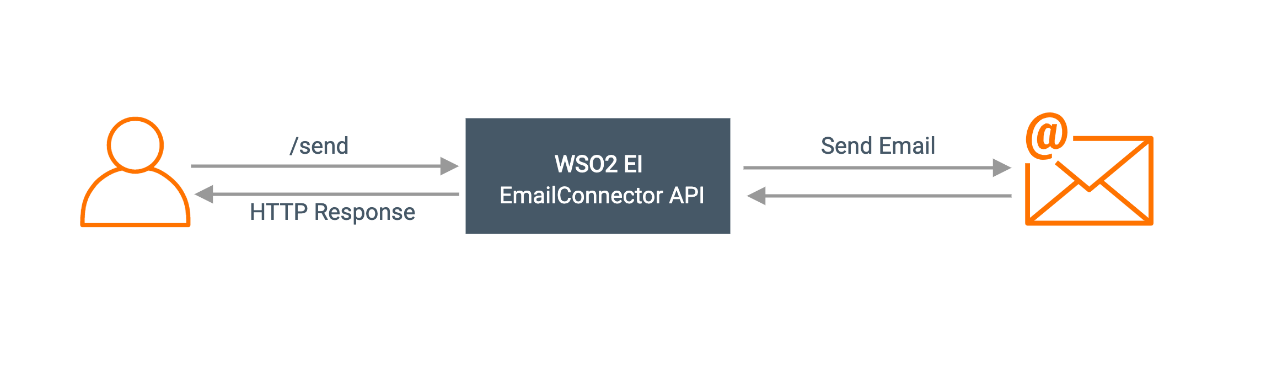
Email Connector can be used to perform operations using protocols SMTP, IMAP and POP3.

What you'll build

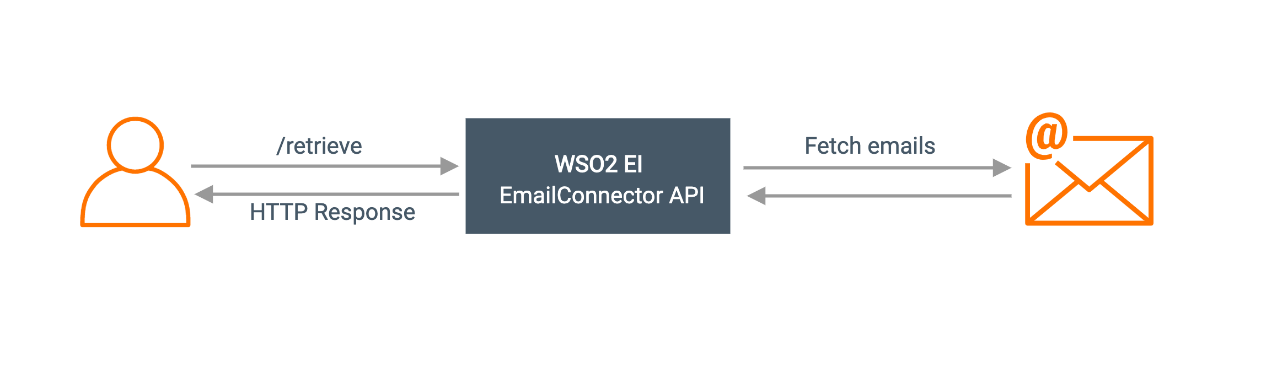
This example explains how to use Email Connector to send an email and retrieve the same email from Gmail. The user sends a payload with the recipients and content of the email. Then, by invoking another API resource, the content of the sent email will be retrieved.

The example consists of an API named as EmailConnector API with two resources send and retrieve.

* /send: The user sends the request payload which includes the recipients, content and attachments of the email. This request is sent to the integration runtime by invoking the EmailConnector API. It will send the email to the relevant recipients.



* /retrieve: The user sends the request payload, containing the filter to search the received email. This request is sent to the integration runtime where the EmailConnector API resides. Once the API is invoked, it returns the filtered emails.

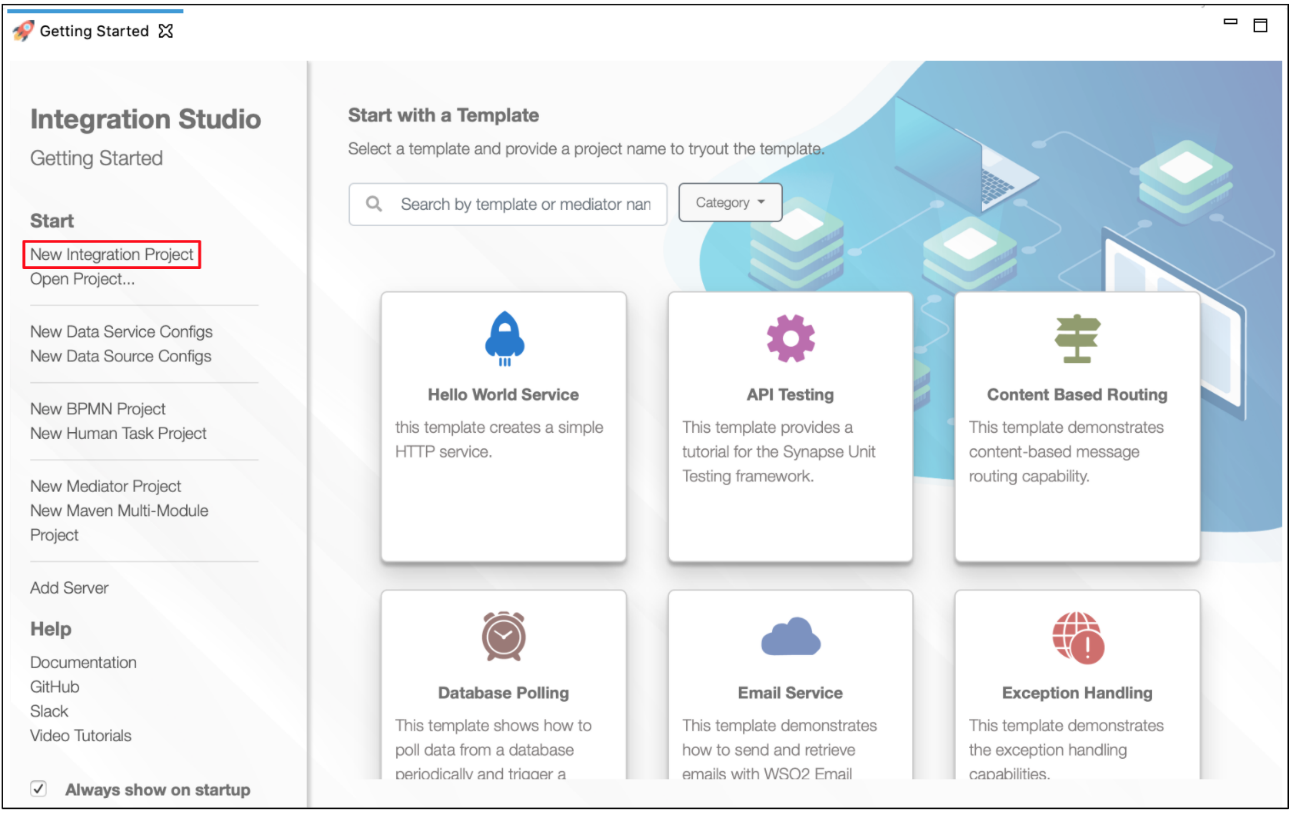


If you do not want to configure this yourself, you can simply [get the project](https://apim.docs.wso2.com/en/latest/reference/connectors/email-connector/email-connector-example/#get-the-project) and run it.

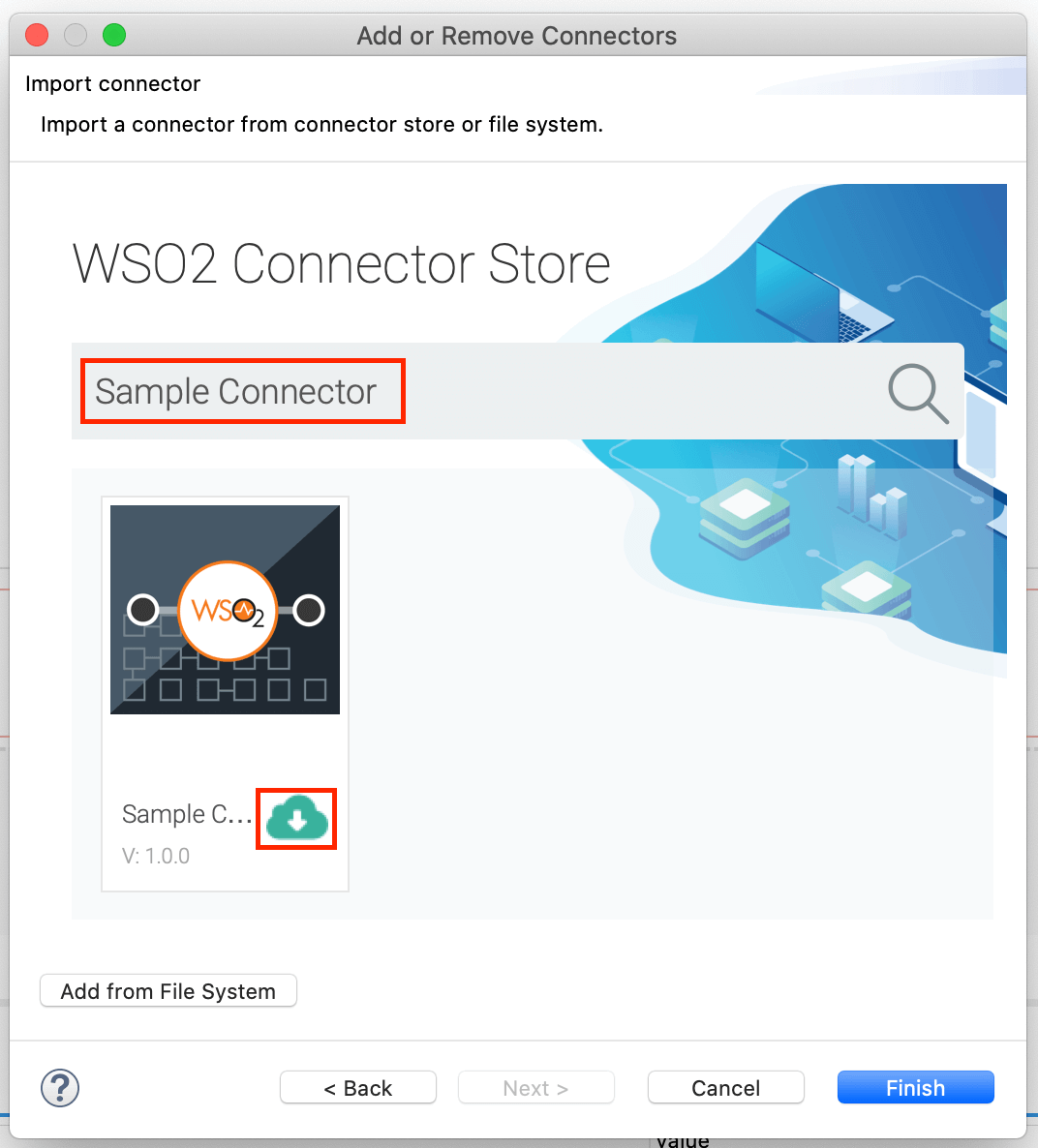
Configure the connector in WSO2 Integration Studio

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

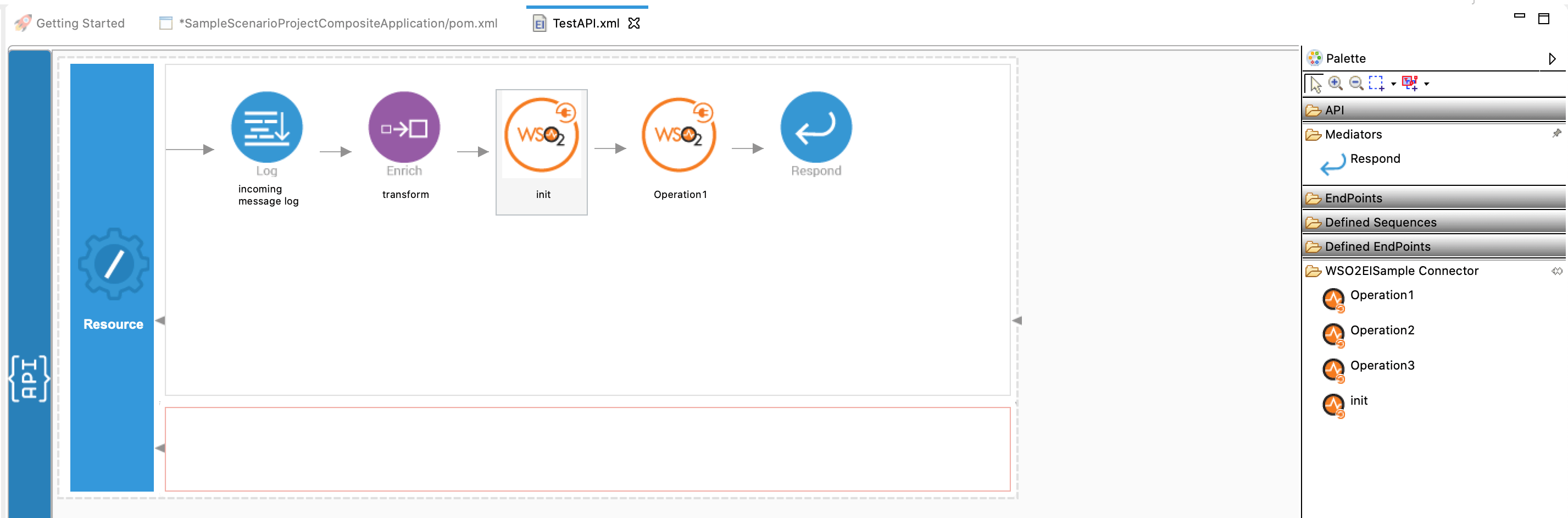
1. Open WSO2 Integration Studio and create an **Integration Project**.

[](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)

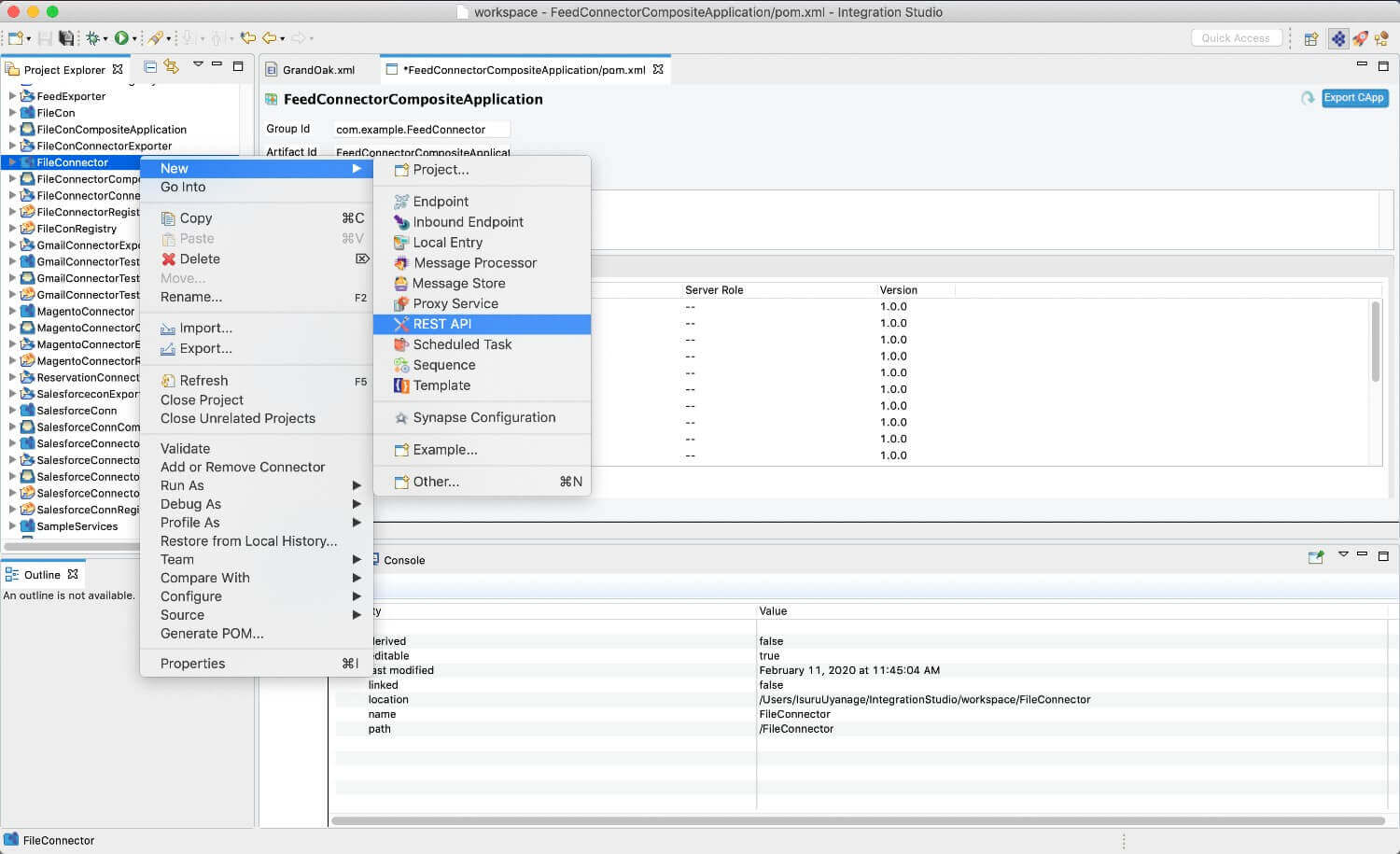
1. Click **Finish**, and your Integration Project is ready. The downloaded connector is displayed on the side palette with its operations.
2. You can drag and drop the operations to the design canvas and build your integration logic.

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

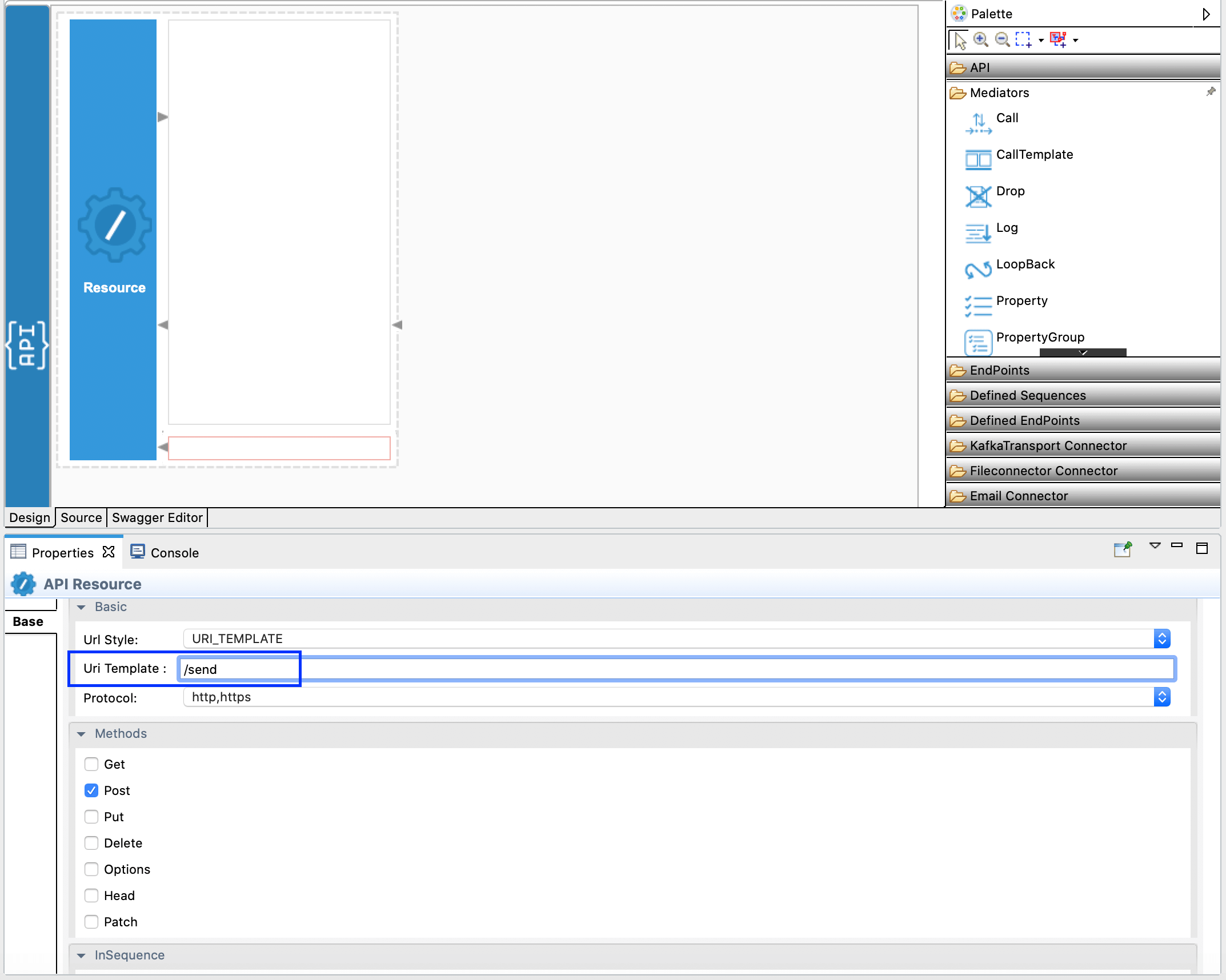
1. Right click on the created Integration Project and select **New** -> **Rest API** to create the REST API.

Creating the Integration Logic

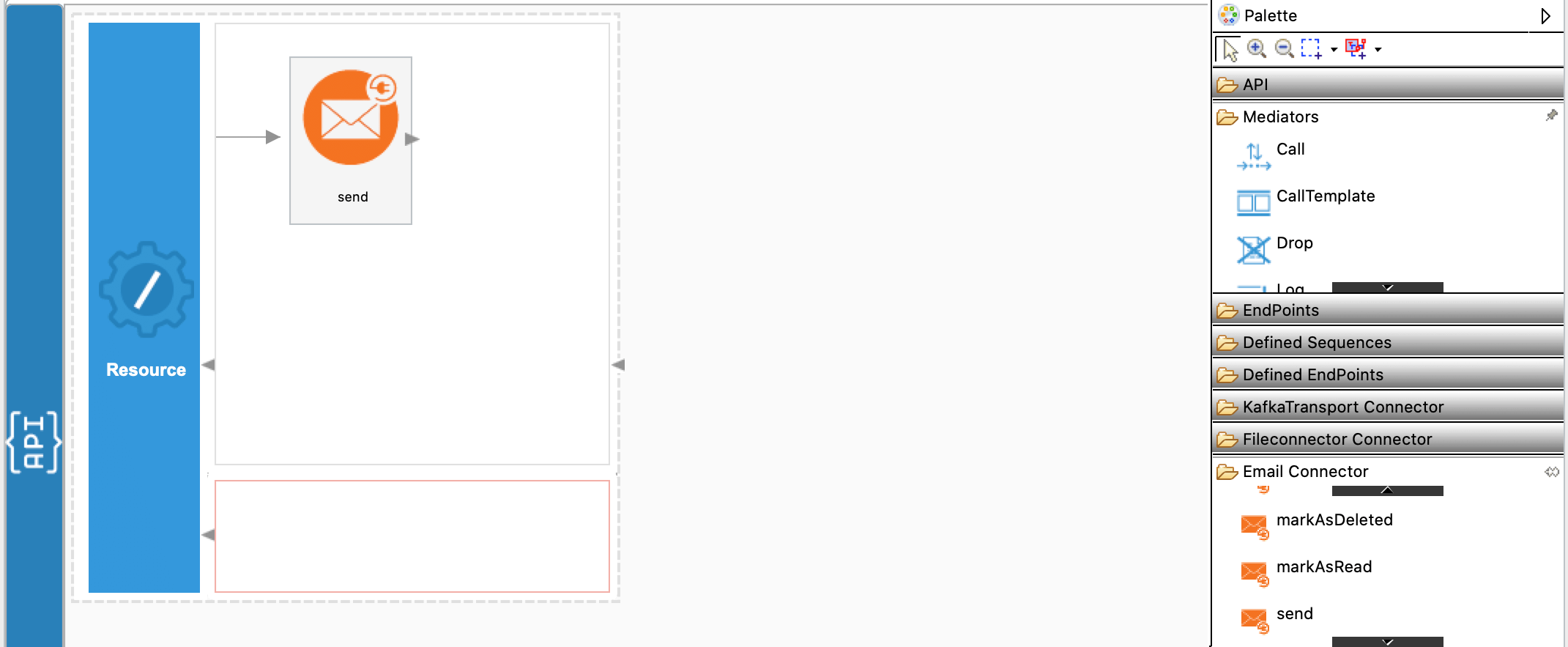
1. Right click on the created Integration Project and select, -> **New** -> **Rest API** to create the REST API.



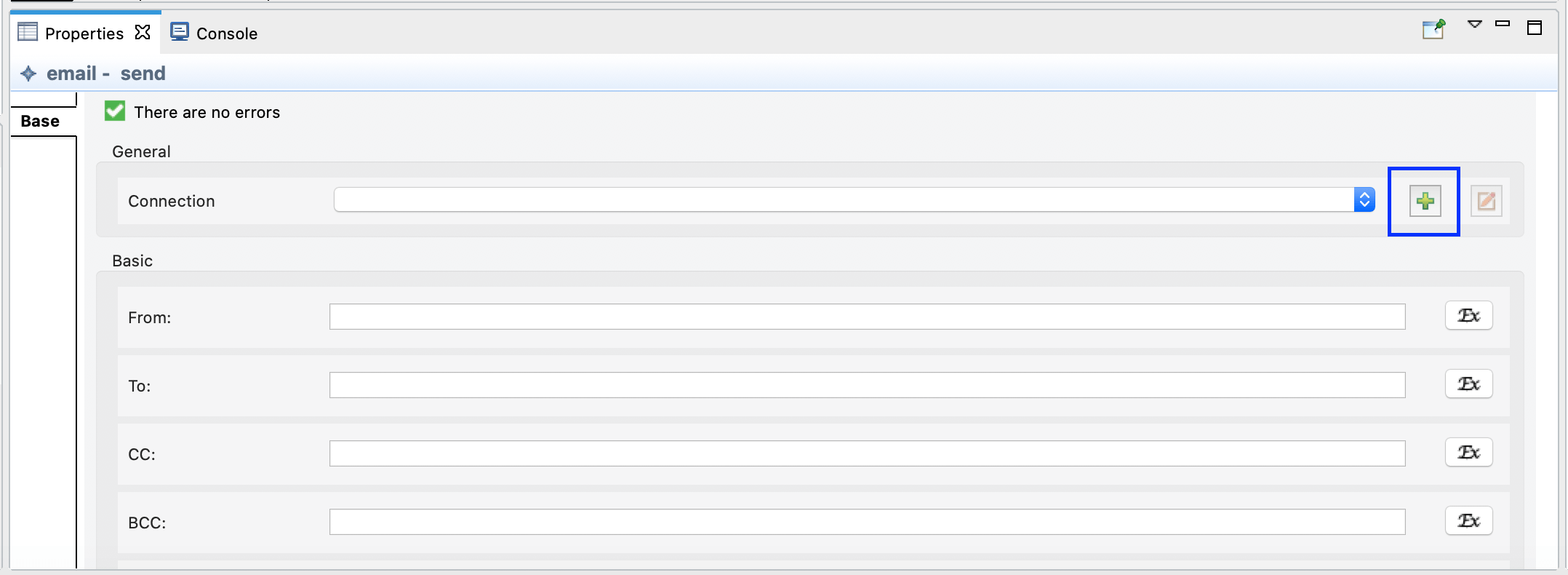
1. Provide the API name as Email Connector and the API context as /emailconnector.
2. First we will create the /send resource. This API resource will retrieve information from the incoming HTTP post request such as recipients and content and construct the email and send to the mentioned recipients.  
   Right click on the API Resource and go to **Properties** view. We use a URL template called /send as we have two API resources inside single API. The method will be Post.



1. In this operation we are going to receive following inputs from the user.
   * from - Sender of the email.
   * to - Recipient of the email.
   * subject - Subject of the email.
   * content - Content to be included in the email.
   * contentType - Content Type of the email
2. Drag and drop the 'send' operation of the Email Connector to the Design View as shown below.



1. Create a connection from the properties window by clicking on the '+' icon as shown below.



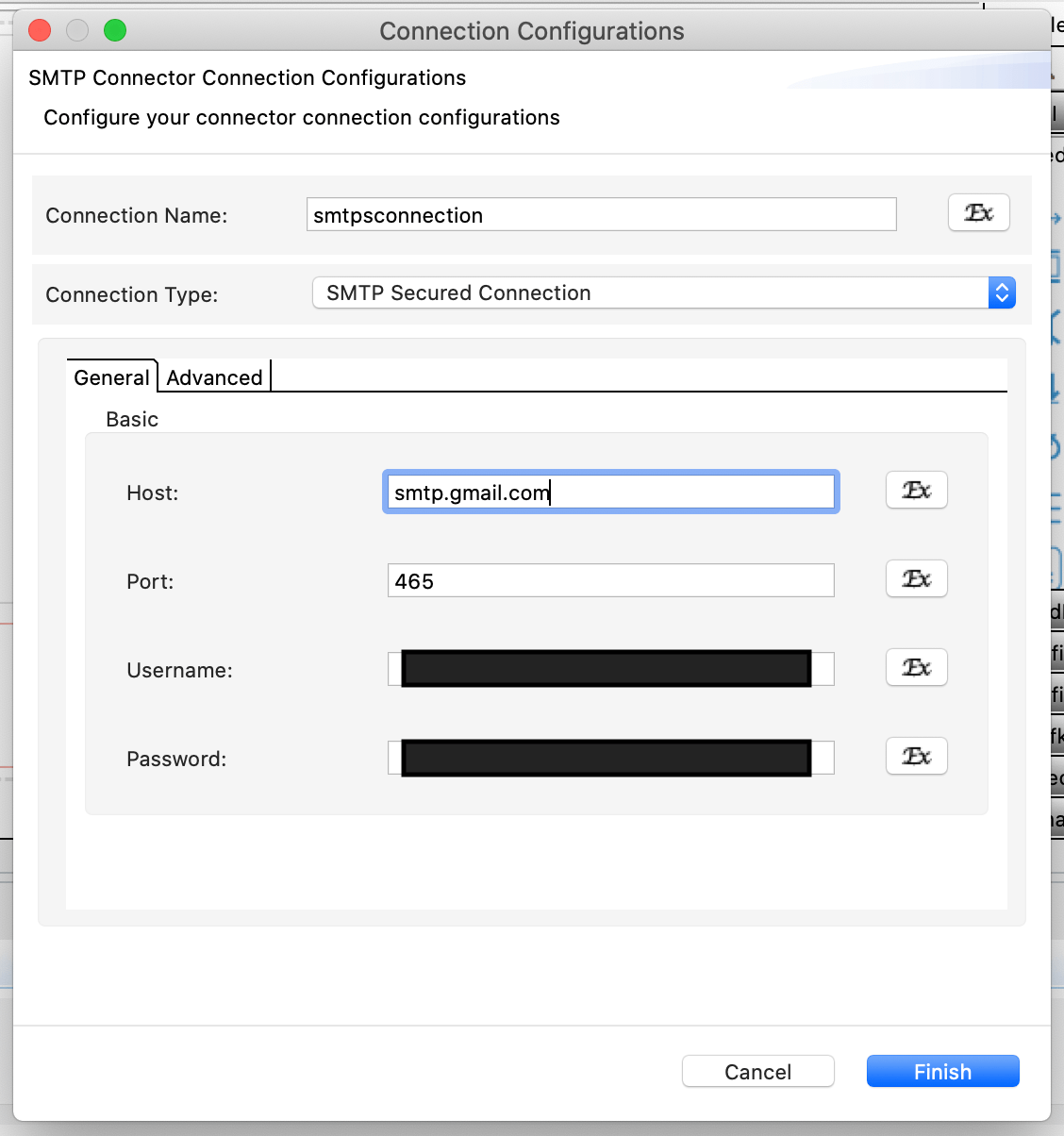
In the pop up window, following parameters must be provided.

* + Connection Name - Unique name to identify the connection by.
  + Connection Type - Type of the connection which specifies the protocol to be used.
  + Host - Host name of the mail server.
  + Port - The port number of the mail server.
  + Username - Username used to connect with the mail server.
  + Password - Password to connect with the mail server.

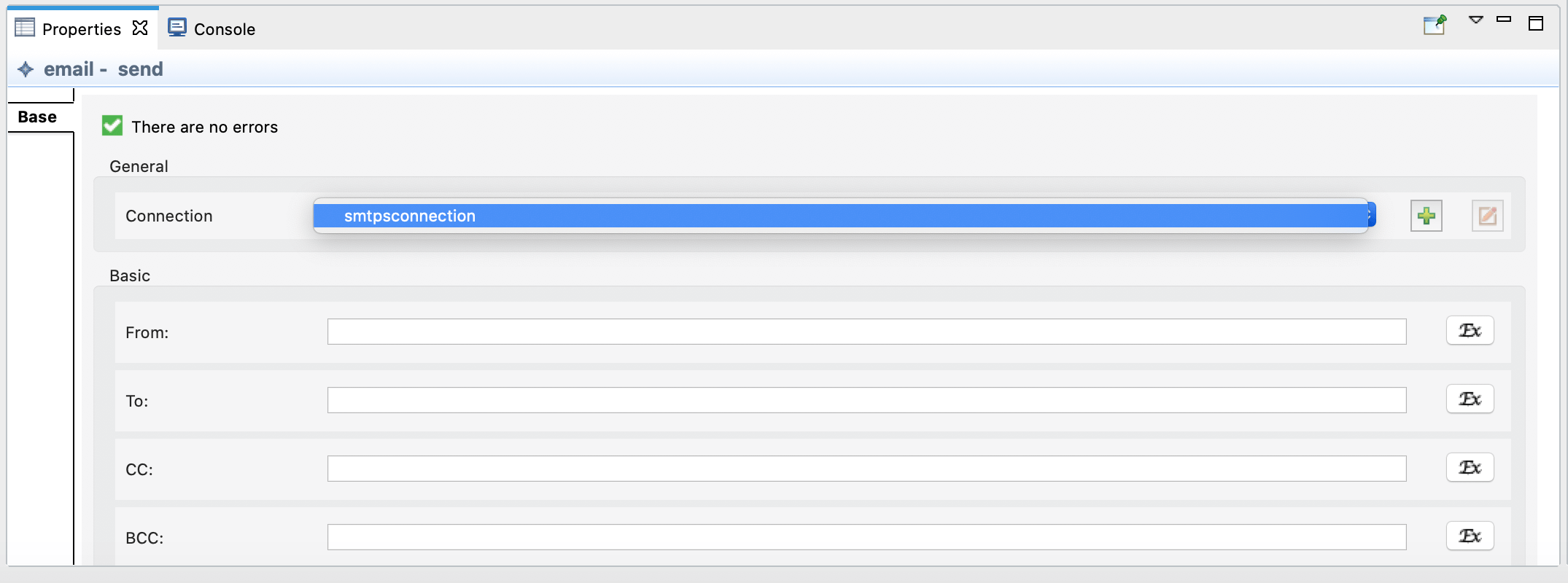
Following values can be provided to connect to Gmail.

* + Connection Name - smtpsconnection
  + Connection Type - SMTP Secured Connection
  + Host - smtp.gmail.com
  + Port - 465
  + Username - <your\_email\_address>
  + Password - <your\_email\_password>

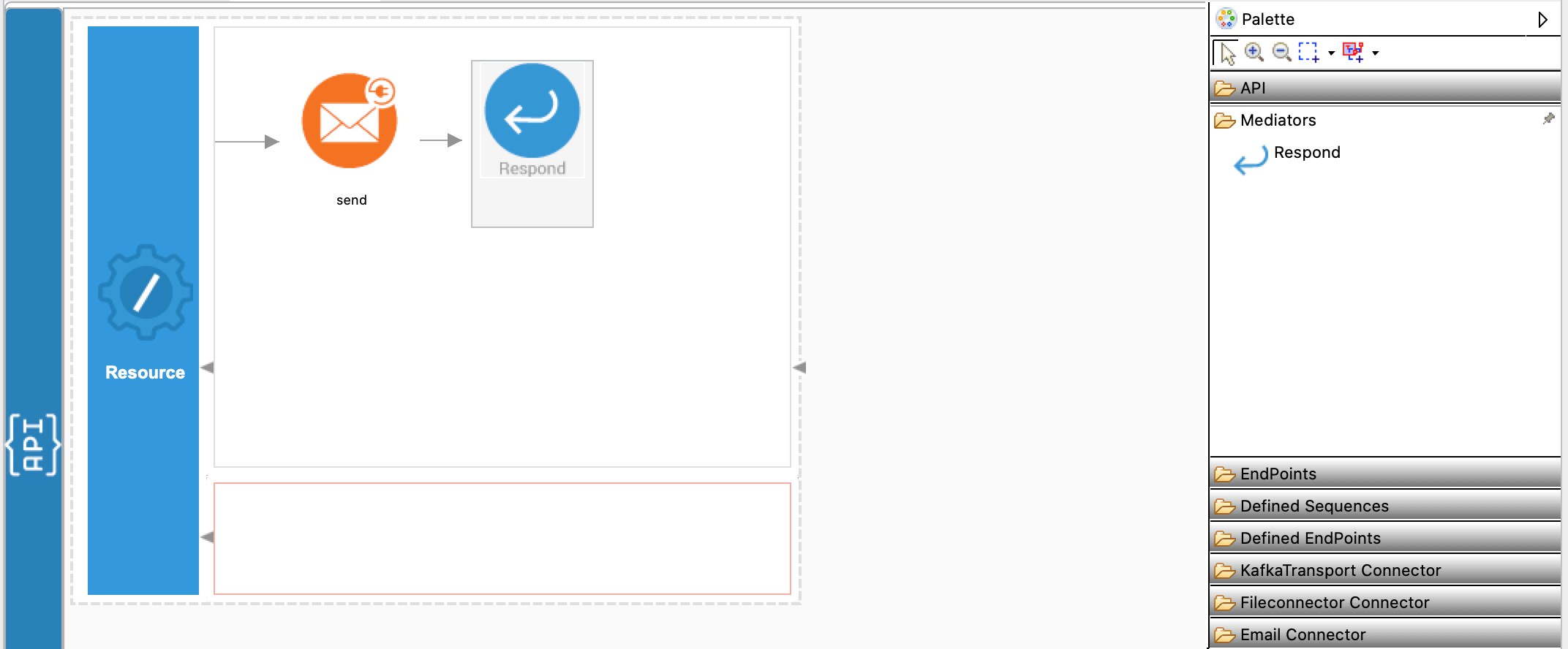
**NOTE**: If you have enabled 2-factor authentication, an app password should be obtained as instructed [here](https://support.google.com/accounts/answer/185833?hl=en).

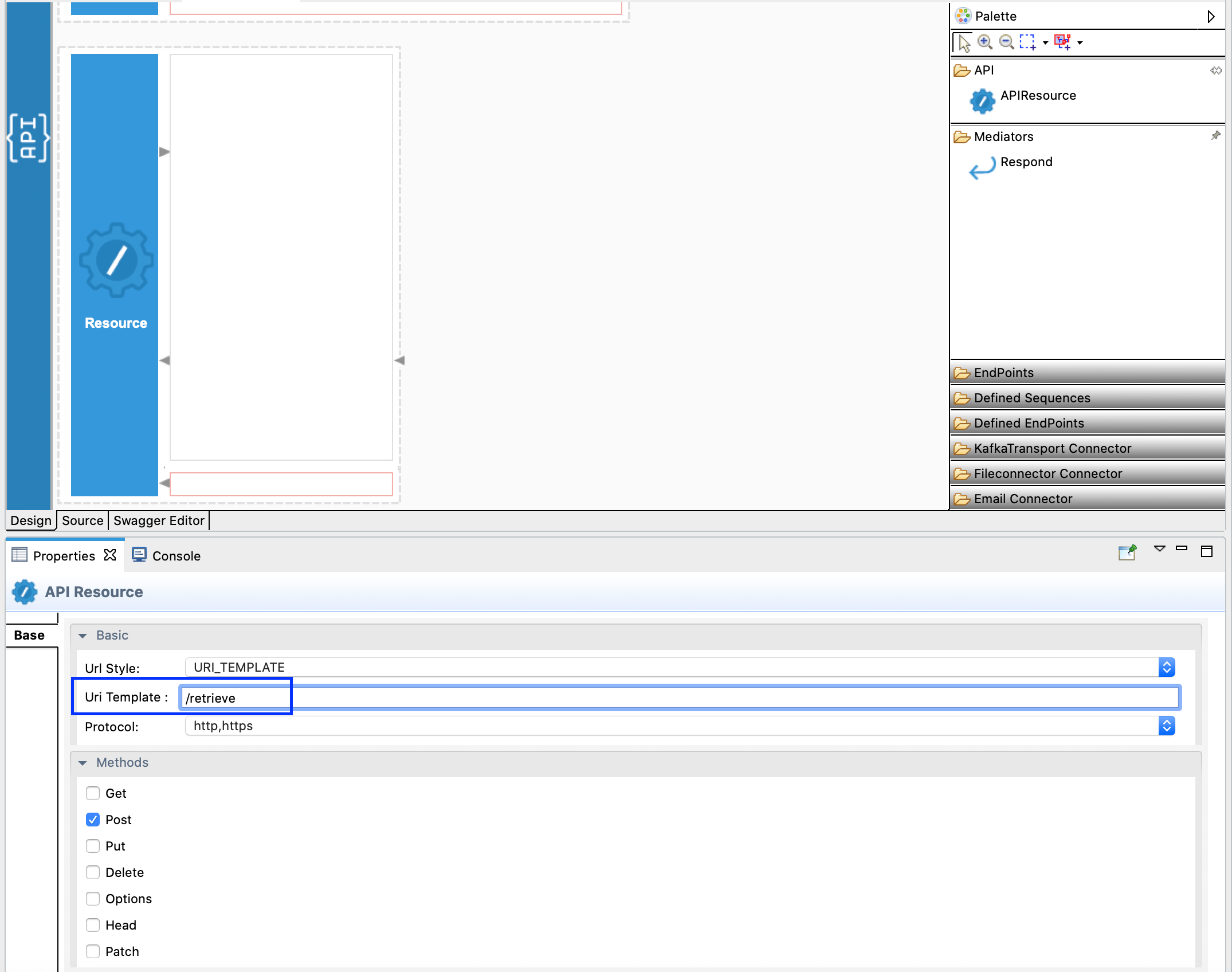


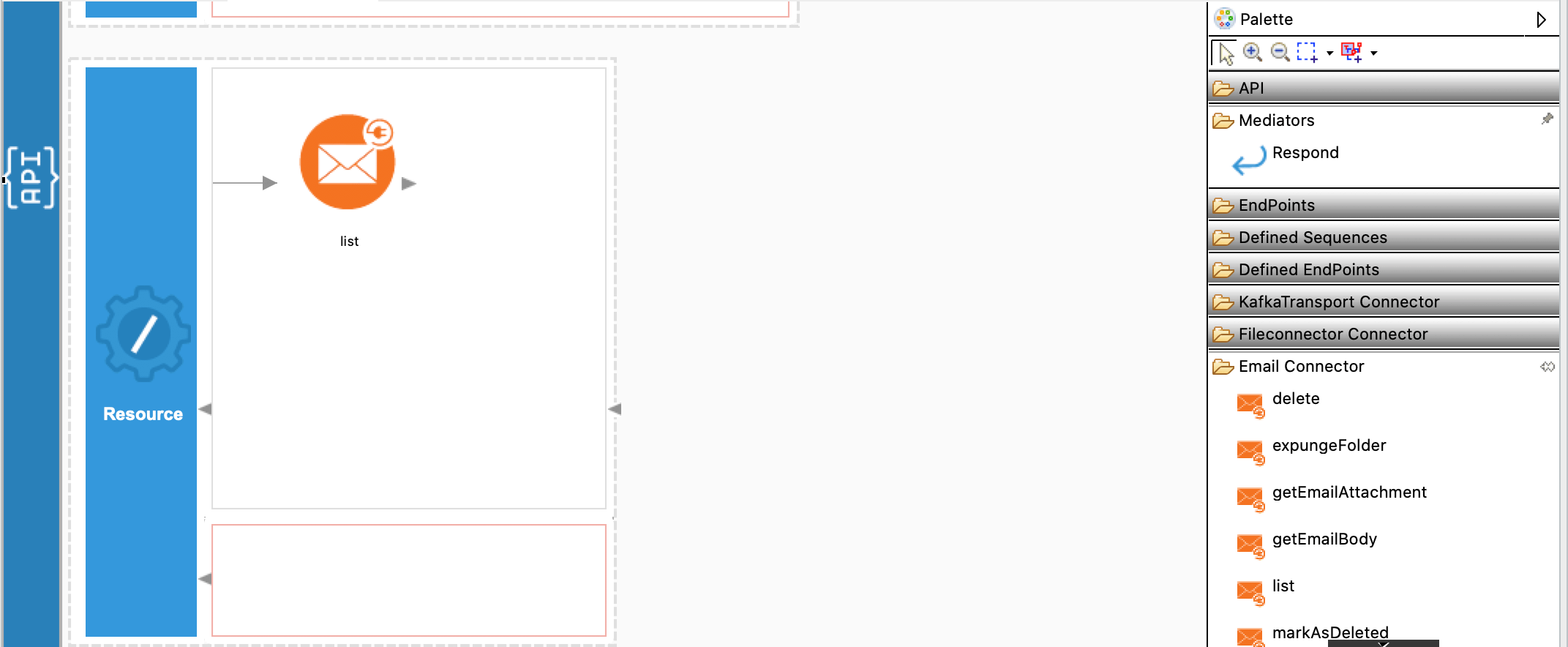
1. After the connection is successfully created, select the created connection as 'Connection' from the drop down in the properties window.



1. Next, provide the expressions as below to the following properties in the properties window to obtain respective values from the JSON request payload.
   * to - json-eval($.to)
   * from - json-eval($.from)
   * subject - json-eval($.subject)
   * content - json-eval($.content)
   * contentType - json-eval($.contentType)
2. Drag and drop the [Respond Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/respond-mediator/) to respond the response from sending the email as shown below.



1. Create the next API resource, which is /retrieve by dragging and dropping another API resource to the design view. This API resource will retrieve filters from the incoming HTTP post request from which to filter the email messages such as the subject, retrieve the emails, retrieve email body and respond back. This will be used to retrieve the email we just sent. This will also be a POST request. 
2. Drag and drop the 'list' operation of the Email Connector to the Design View as shown below.

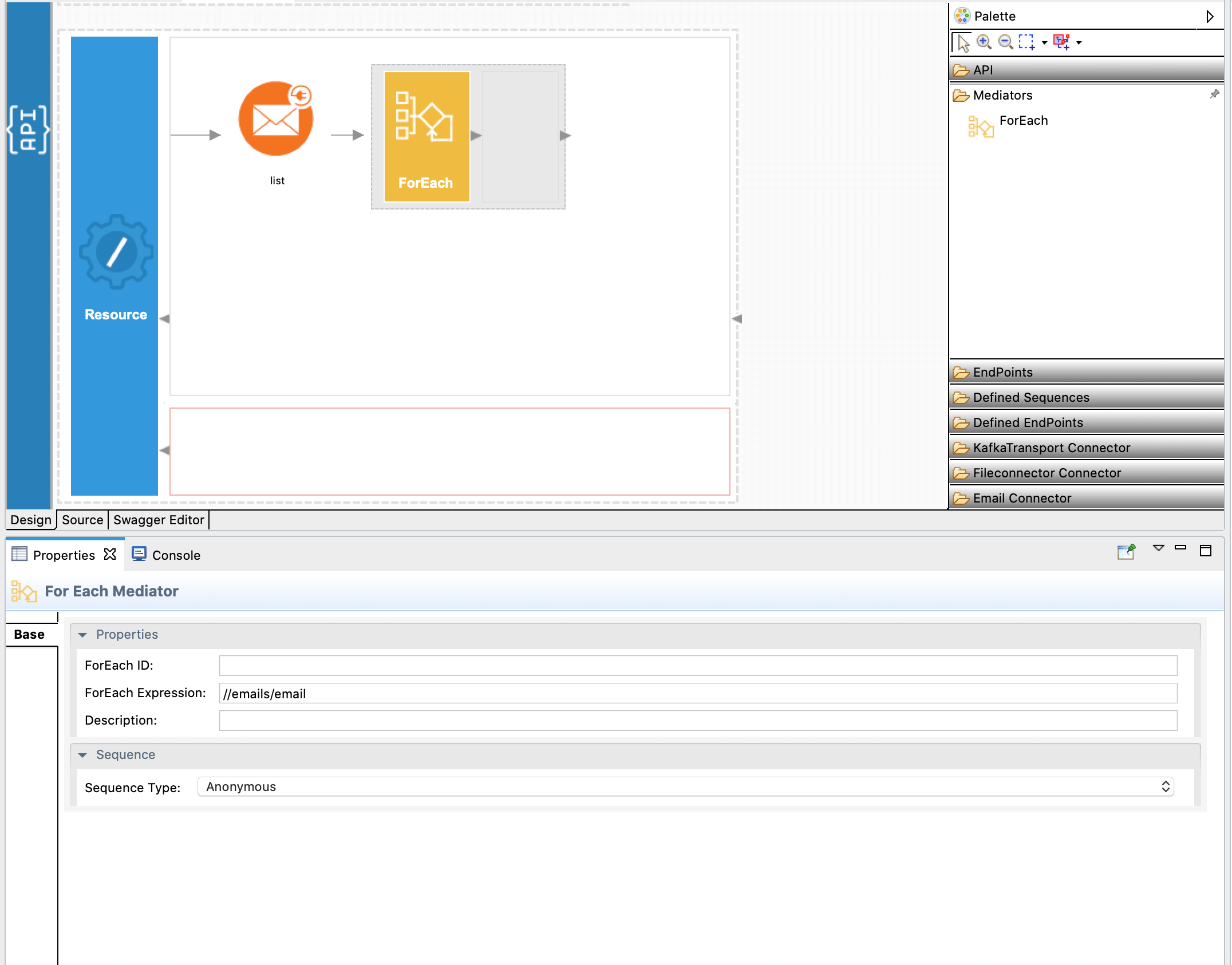


1. Next, we will create a IMAP connection to list emails similar to step 6. Following are the values to be provided when creating the connection.
   * Connection Name - imapsconnection
   * Connection Type - IMAP Secured Connection
   * Host - imap.gmail.com
   * Port - 993
   * Username - <your\_email\_address>
   * Password - <your\_email\_password>
2. In this operation we are going to receive following inputs from the user.
   * subjectRegex - Subject Regex to filter the email from.

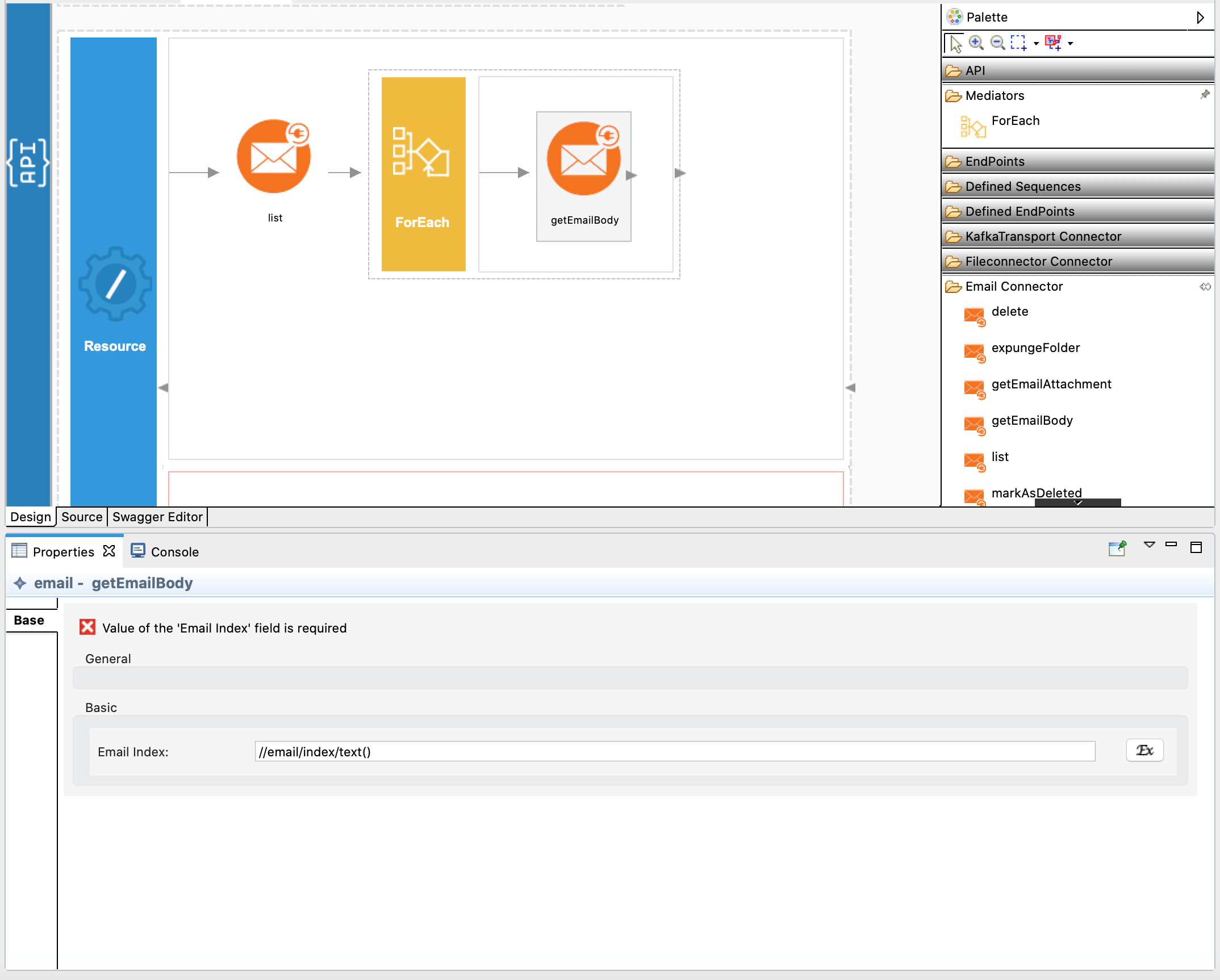
Therefore, provide the expressions as below to the following properties in the properties window to obtain respective values from the JSON request payload.

* + Subject Regex: json-eval($.subjectRegex)

1. We will next iterate the response received and obtain the email content of each email using the getEmailBody operation. In order to do this, drag and drop the [Foreach Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/foreach-mediator/) as shown below and enter //emails/email as the Foreach Expression in the properties window.

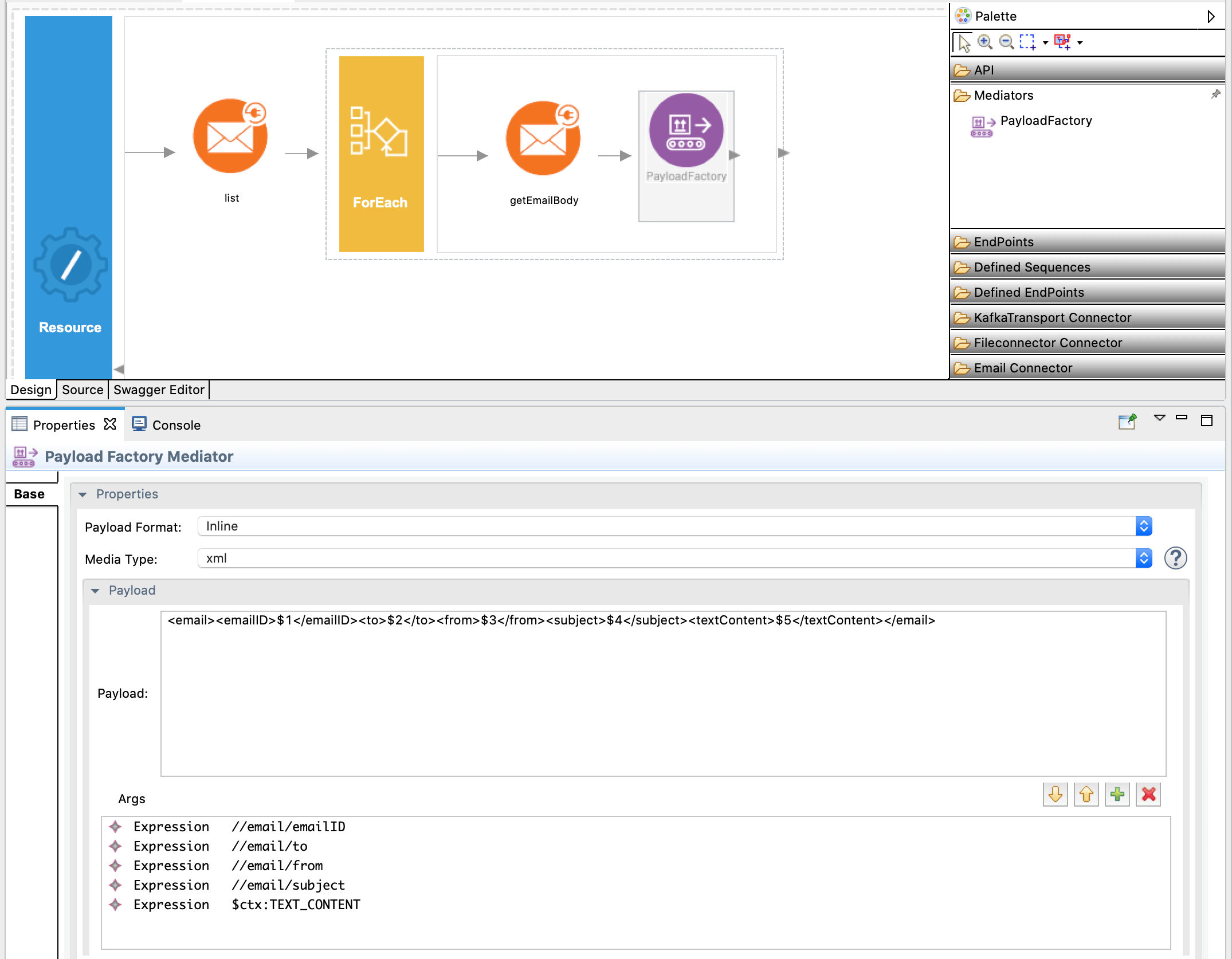


1. Inside the [Foreach Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/foreach-mediator/), drag and drop the getEmailBody operation as shown below and provide the //email/index/text() expression as the Email Index.



**NOTE**: Further, you can use getAttachment operation to retrieve attachment content if there are any. Refer [Reference Documentation](https://apim.docs.wso2.com/en/latest/reference/connectors/email-connector/email-connector-example/email-connector-config/) to learn more.

1. Next, we will use a [Payload Factory Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/payloadfactory-mediator/), to add the email content to the same response we received from list operation and configure the Payload mediator as shown below.



Enter following as the payload:

<**email**>

<**emailID**>$1</**emailID**>

<**to**>$2</**to**>

<**from**>$3</**from**>

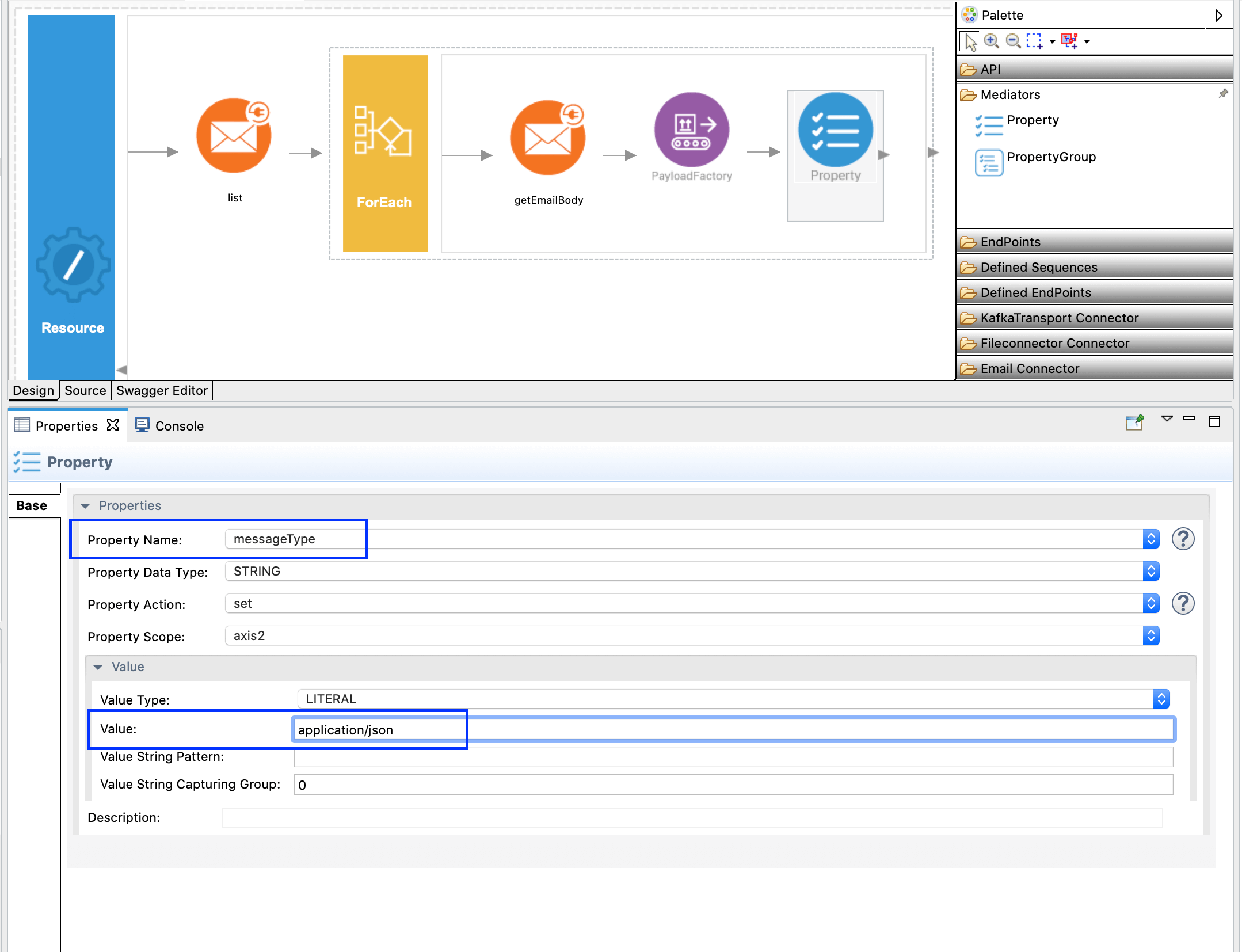
<**subject**>$4</**subject**>

<**textContent**>$5</**textContent**>

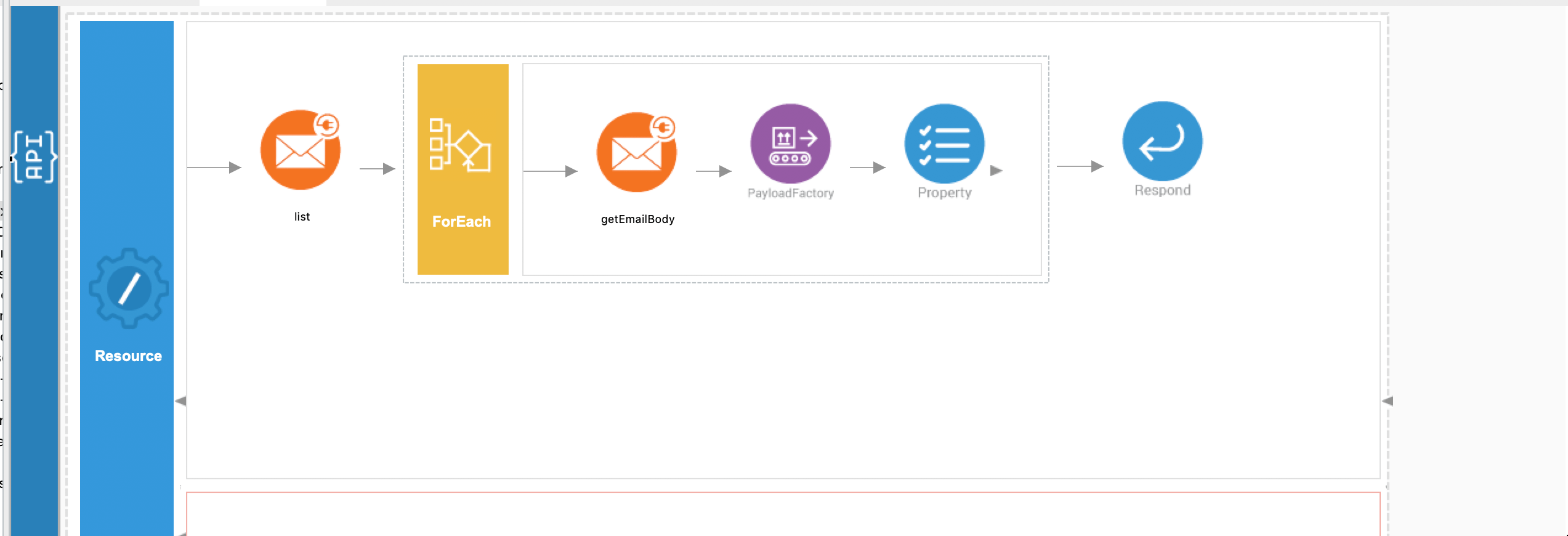
</**email**>

Here, you may observe that we are obtaining TEXT\_CONTENT property which is being set when getEmailBody is invoked to retrieve the email content. You can find the list of similar properties set in this operation [here](https://apim.docs.wso2.com/en/4.1.0/reference/connectors/email-connector/email-connector-config/).

1. Drag and drop a [Property Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/property-mediator/) and set the Property name as 'messageType' and the value as application/json. This is added so that the response will be in json.



1. Finally, drag and drop the [Respond Mediator](https://apim.docs.wso2.com/en/4.1.0/reference/mediators/respond-mediator/) after the 'foreach' mediator to respond the response of retrieved emails.



1. You can find the complete API XML configuration below. You can go to the source view and copy paste the following config.

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

<**api** context="/emailconnector" name="EmailConnector" xmlns="http://ws.apache.org/ns/synapse">

<**resource** methods="POST" uri-template="/send">

<**inSequence**>

<**email.send** configKey="smtpsconnection">

<**from**>{json-eval($.from)}</**from**>

<**to**>{json-eval($.to)}</**to**>

<**subject**>{json-eval($.subject)}</**subject**>

<**content**>{json-eval($.content)}</**content**>

<**contentType**>{json-eval($.contentType)}</**contentType**>

</**email.send**>

<**respond**/>

</**inSequence**>

<**outSequence**/>

<**faultSequence**/>

</**resource**>

<**resource** methods="POST" uri-template="/retrieve">

<**inSequence**>

<**email.list** configKey="imapsconnection">

<**subjectRegex**>{json-eval($.subjectRegex)}</**subjectRegex**>

</**email.list**>

<**foreach** expression="//emails/email">

<**sequence**>

<**email.getEmailBody**>

<**emailIndex**>{//email/index/text()}</**emailIndex**>

</**email.getEmailBody**>

<**payloadFactory** media-type="xml">

<**format**>

<**email** xmlns="">

<**emailID**>$1</**emailID**>

<**to**>$2</**to**>

<**from**>$3</**from**>

<**subject**>$4</**subject**>

<**textContent**>$5</**textContent**>

</**email**>

</**format**>

<**args**>

<**arg** evaluator="xml" expression="//email/emailID"/>

<**arg** evaluator="xml" expression="//email/to"/>

<**arg** evaluator="xml" expression="//email/from"/>

<**arg** evaluator="xml" expression="//email/subject"/>

<**arg** evaluator="xml" expression="$ctx:TEXT\_CONTENT"/>

</**args**>

</**payloadFactory**>

</**sequence**>

</**foreach**>

<**property** name="messageType" scope="axis2" type="STRING" value="application/json"/>

<**respond**/>

</**inSequence**>

<**outSequence**/>

<**faultSequence**/>

</**resource**>

</**api**>

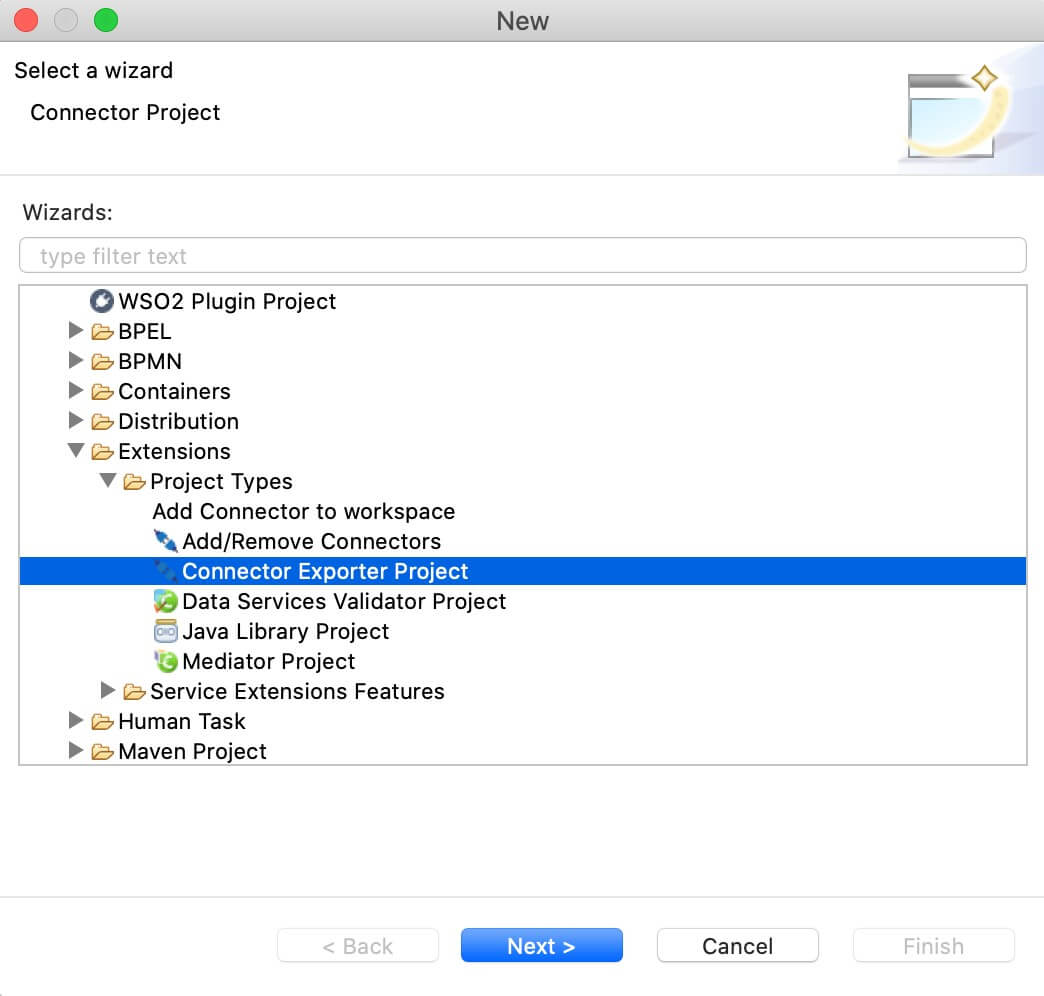
Exporting Integration Logic as a CApp

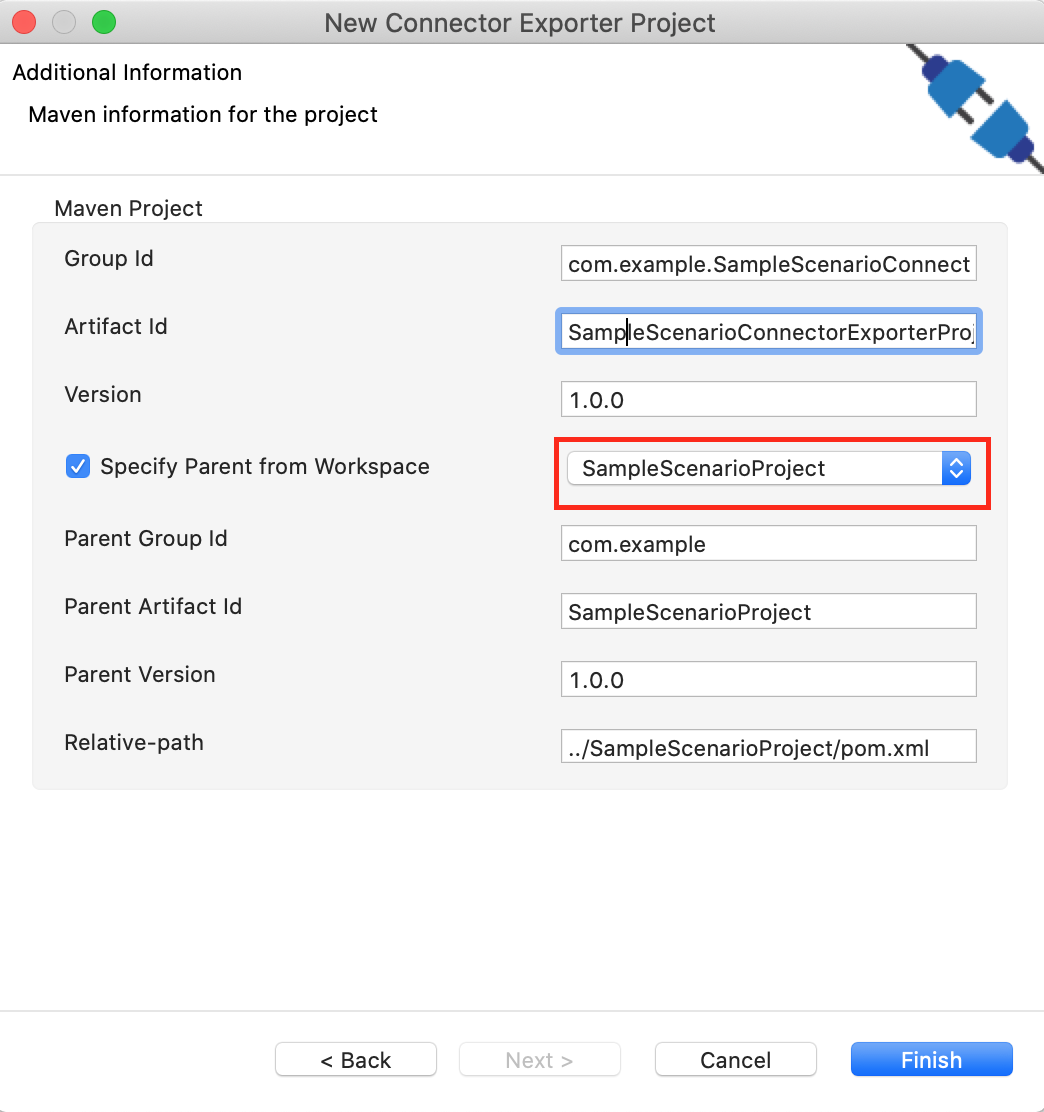
**CApp (Carbon Application)** is the deployable artifact on the integration runtime. Let us see how we can export integration logic we developed into a CApp along with the connector.

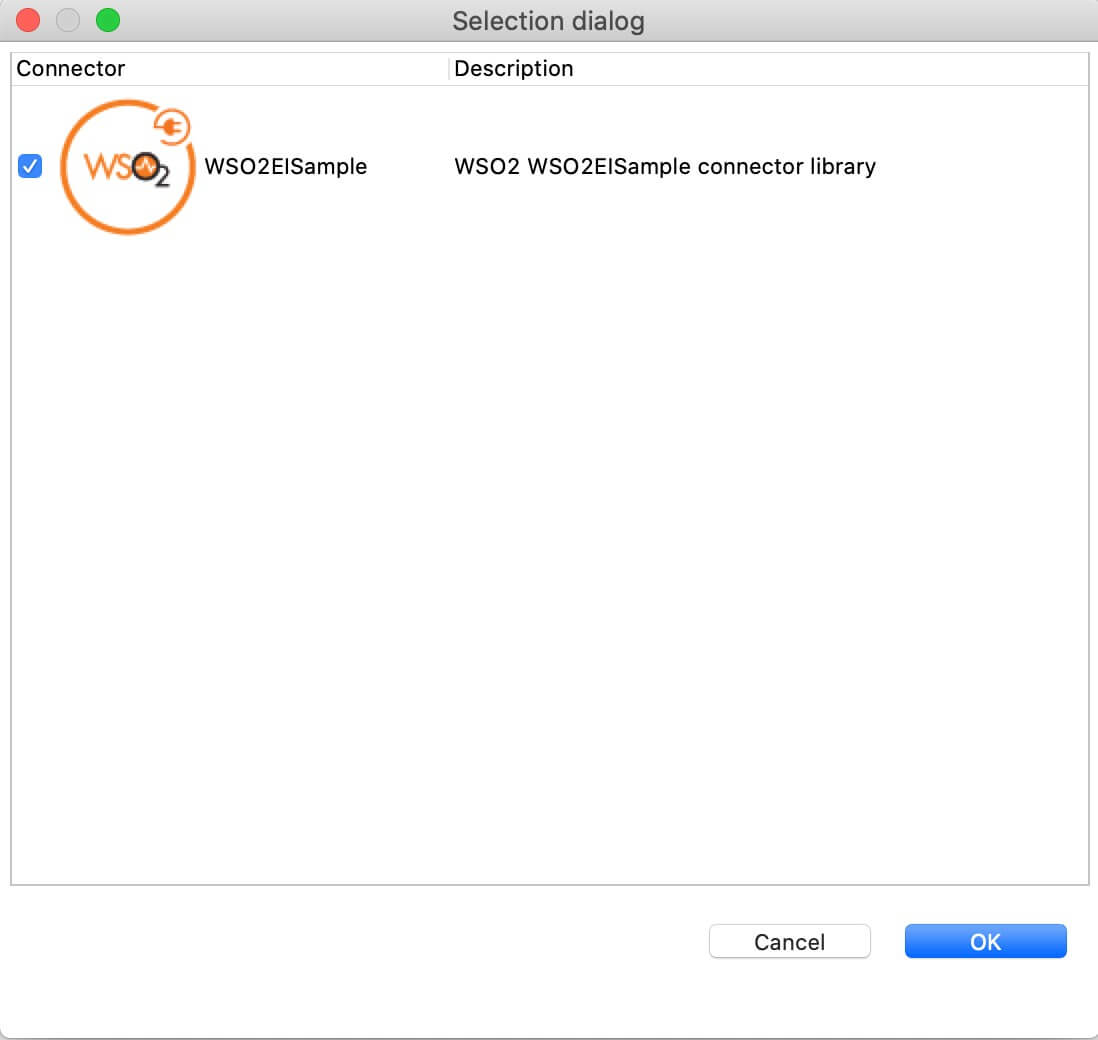
Creating Connector Exporter Project

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**.

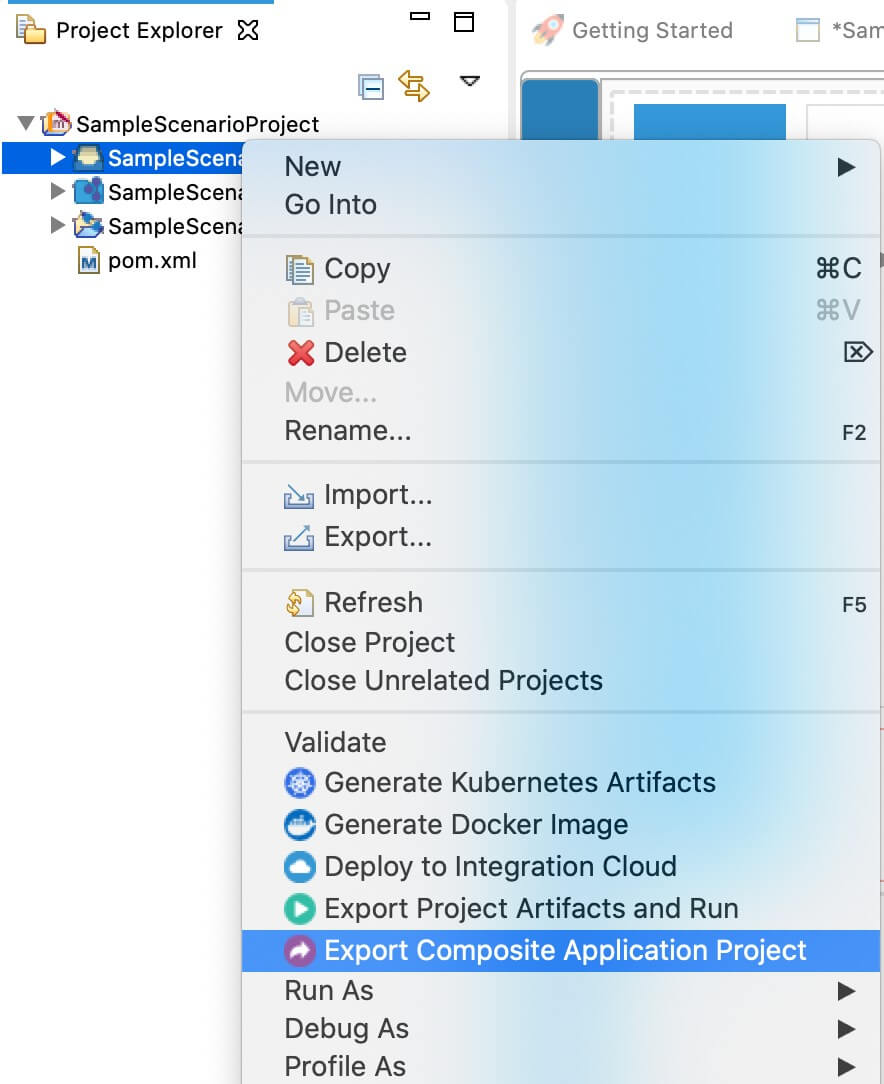
[](https://apim.docs.wso2.com/en/4.1.0/assets/img/integrate/connectors/adding-connector-to-exporter-project-3.jpg)

[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)

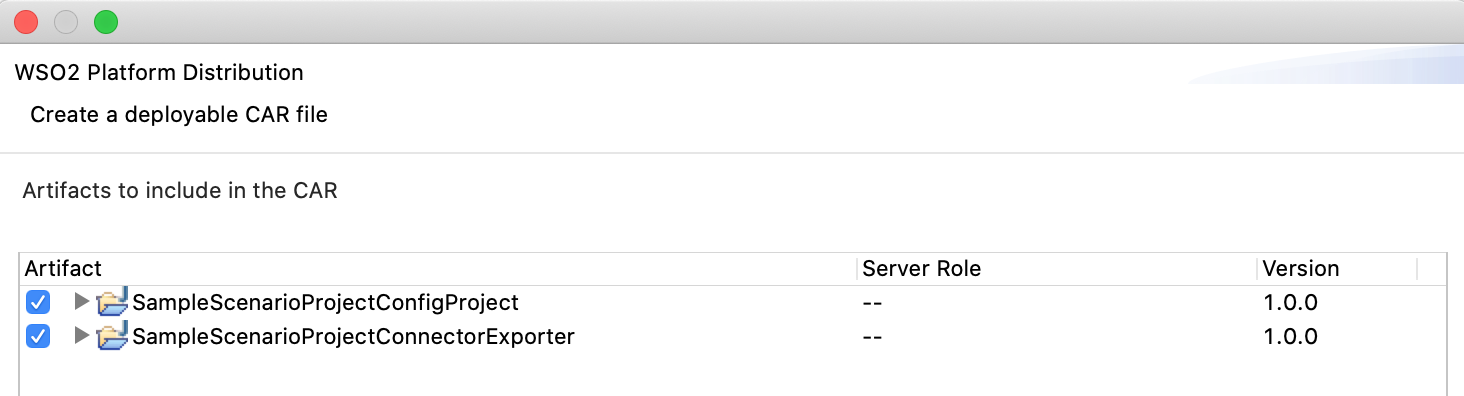
To export the Integration Project as a CApp, a Composite Application Project needs to be created. Usually, when an Integration project is created, this project can be created as part of that project by Integration Studio. If not, you can specifically create it by navigating to **File** -> **New** ->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

Get the project

You can download the ZIP file and extract the contents to get the project code.

[](https://apim.docs.wso2.com/en/4.1.0/assets/attachments/connectors/emailconnector.zip)

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

Email Send Operation

1. Create a file called data.json with the following payload. We will send the email to ourself so that we can retrieve it later.

{

"from": "<your-email>@gmail.com",

"to": "<your-email>@gmail.com",

"subject": "Sample email",

"content": "This is the body of the sample email",

"contentType":"text/plain"

}

1. Invoke the API as shown below using the curl command. Curl Application can be downloaded from [here](https://curl.haxx.se/download.html).

curl -H "Content-Type: application/json" --request POST --data @body.json http://localhost:8290/emailconnector/send

**Expected Response**: You should get a 'Success' response as below and you will receive the email.

{

"result": {

"success": true

}

}

Email List Operation

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

{

"subjectRegex":"Sample email"

}

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

curl -H "Content-Type: application/json" --request POST --data @body.json http://localhost:8290/emailconnector/retrieve

**Expected Response**: You should get a response like below.

{

"emails": {

"email": [

{

"index": 0,

"emailID": "<1623446944.0.152334336343@localhost>",

"to": "<your-email>@gmail.com",

"from": "<your-email>@gmail.com",

"subject": "Sample email",

"textContent": "This is the body of the sample email"

}

]

}

}

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 [Email Connector Configuration](https://apim.docs.wso2.com/en/4.1.0/reference/connectors/email-connector/email-connector-config/) documentation for all operation details of the connector.