# SAP Machine Learning: Image and Product Classification

In this project you will create an app SAP’s trial machine learning services for image and product classification.

### Prerequisites

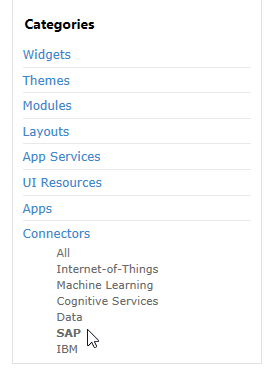
This case was prepared using version 7.21.0 of the desktop Mendix Modeler. You must also have an SAP p or s account.

## Create the App

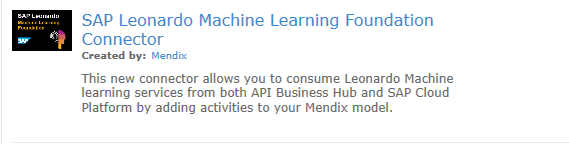
Create an app using the Fiori Blank SAP template. Name the app according to the format provided by your instructor.

## Add the SAP Leonardo Machine Learning Foundation Connector

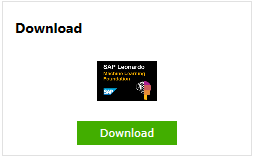
In the Mendix Modeler, click on the shopping cart icon on the upper right. This open the Mendix App Store. Click on Connectors and SAP in the Categories list on the top-left.



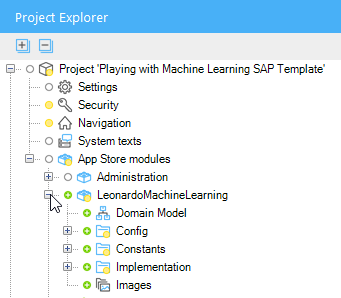
Click SAP Leonard Machine Learning Foundation Connector in the list that opens.



Click Download. Confirm the dialog that opens asking whether you want to add the module to your project.

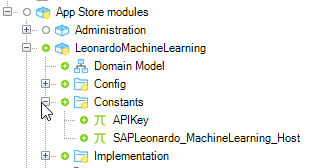


The module is added to the project in the App Store modules folder which you can find inside the top folder. The module adds the domain model entities, constants, microflows, etc. required to access the machine learning services. You can find the Mendix documentation for this module at <https://docs.mendix.com/refguide/sap/sap-leonardo-connector>.

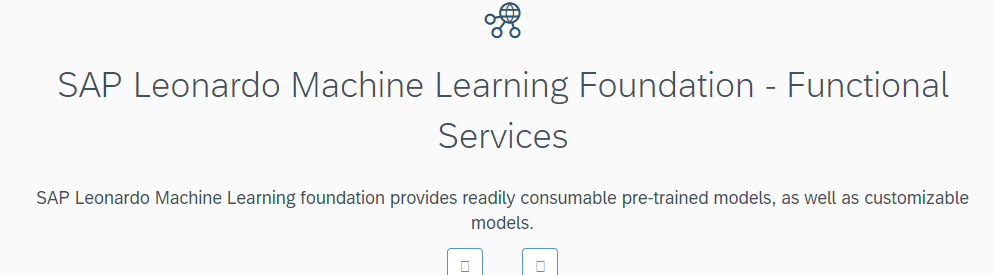


## Configure the API Key

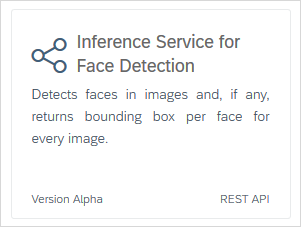
To access the services, you must configure an API key in the APIKey constant.



To obtain a key, navigate to <https://api.sap.com/package/SAPLeonardoMLFunctionalServices?section=Artifacts> .



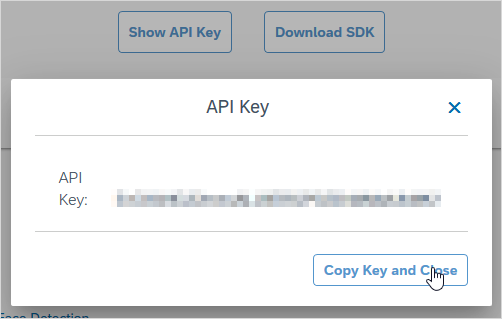
Click on the tile for one of the services. It doesn’t matter which one.



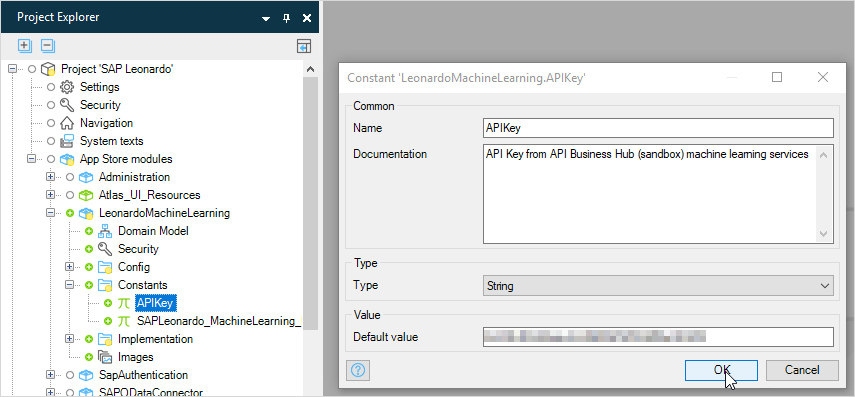
Click Show API Key. If you aren’t logged in with an SAP id, you will be prompted to log in.



Copy the API Key.



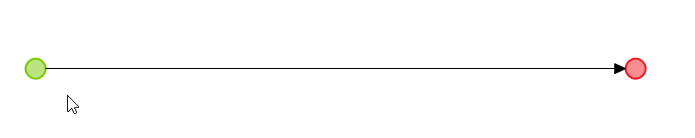
In the Mendix Modeler, open the APIKey constant and paste the API key into the Default value field.



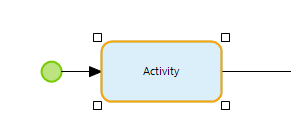
## Create a Microflow to Open the Home Page

If you run the app now, the app opens to the Home page. That is what want to happen but we need to create an image object to hold the image we want to classify first so, in this section we’ll create a microflow that creates the image object, opens the Home page and passes the image object to the page.

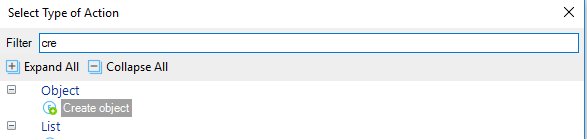
Create a folder in MyFirstModule called Microflows by right-clicking the MyFirstModule module in the Project Explorer and selecting Add folder. Add a microflow called ACT\_OpenHomePage in the Microflows folder. Initially, the microflow has only Start and End events.



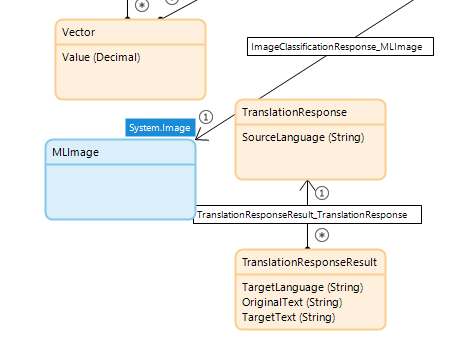
Add an activity to the right of the Start event either by right-clicking the line or dragging from the toolbar above the editor.



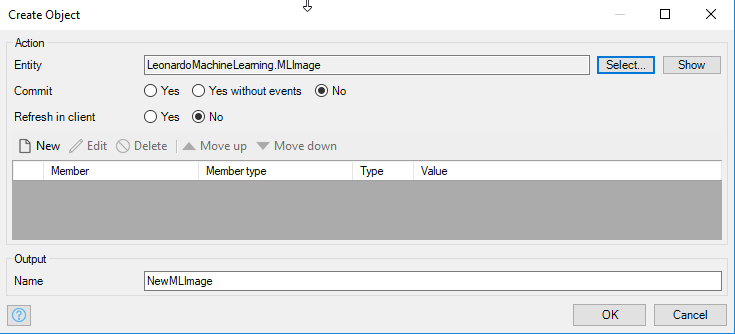
Double-click the activity to select the type and select the Create object type.



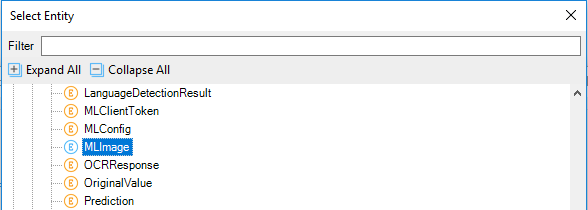
The SAP Leonardo Machine Learning plugin includes a Domain model that has an image entity we can use.



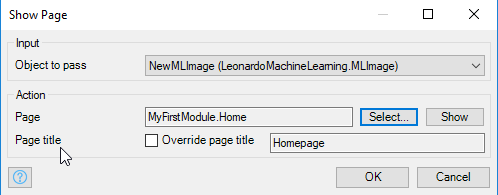
Configure the Create object activity as shown below.



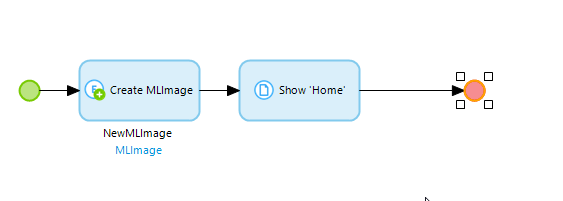
You can find the MLImage Entity in the Domain model of the LeonardoMachineLearning module.



Now add a Show page activity and configure it as shown:

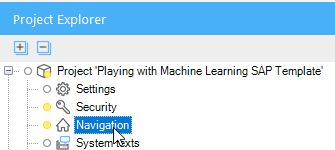


The final microflow looks like this:

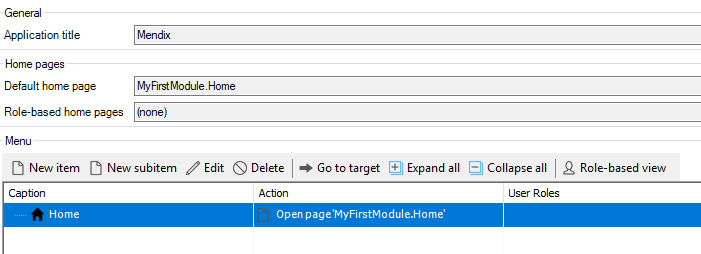


## Configure Navigation to use the Microflow

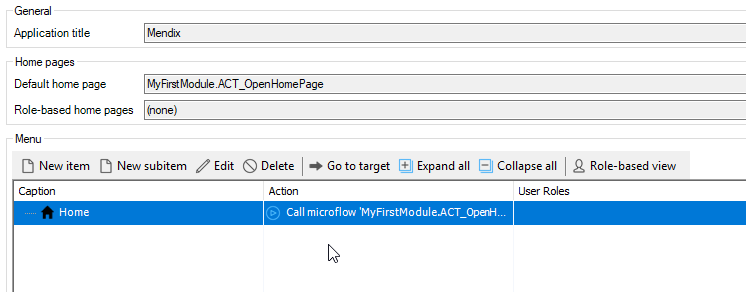
Open Navigation located under the project name in the Project Folder.



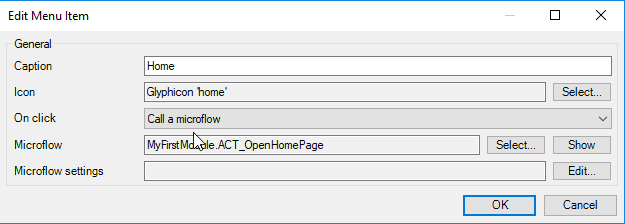
The navigation looks like the image below. We have to update the Default home page field and the Home menu item.



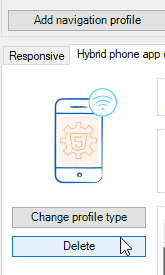
Configure these as shown below.



When you configure the Home menu item, you have to change On click action from Show a page to Call a microflow as shown below.

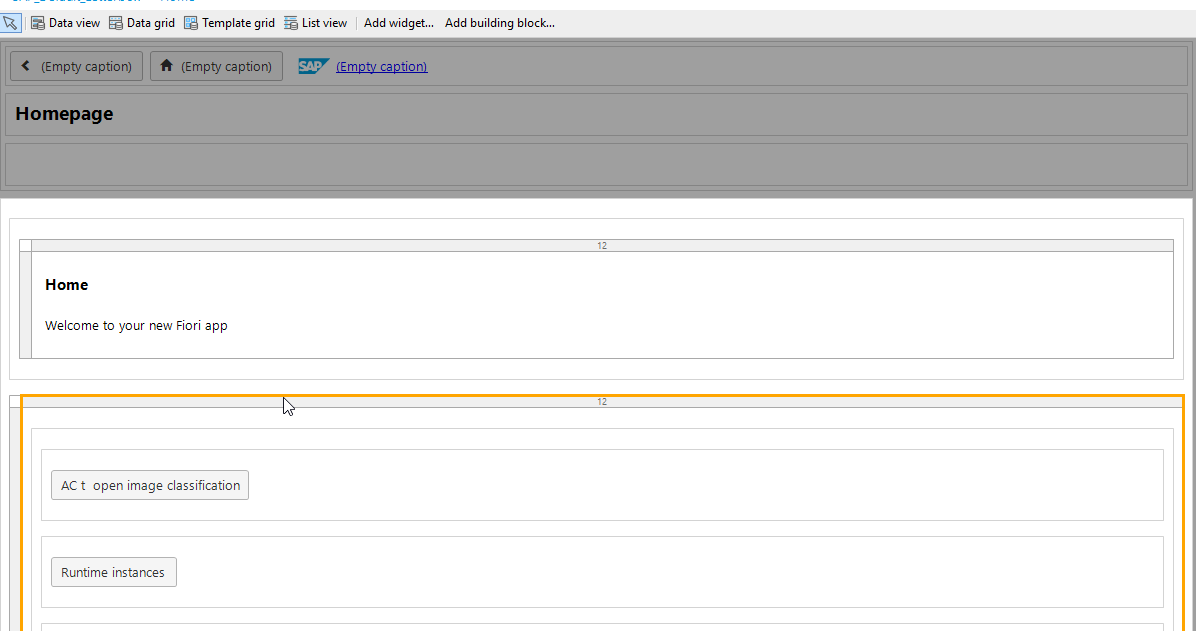


Now select the Hybrid phone app navigation profile on the left side of the Navigation editor and delete it.

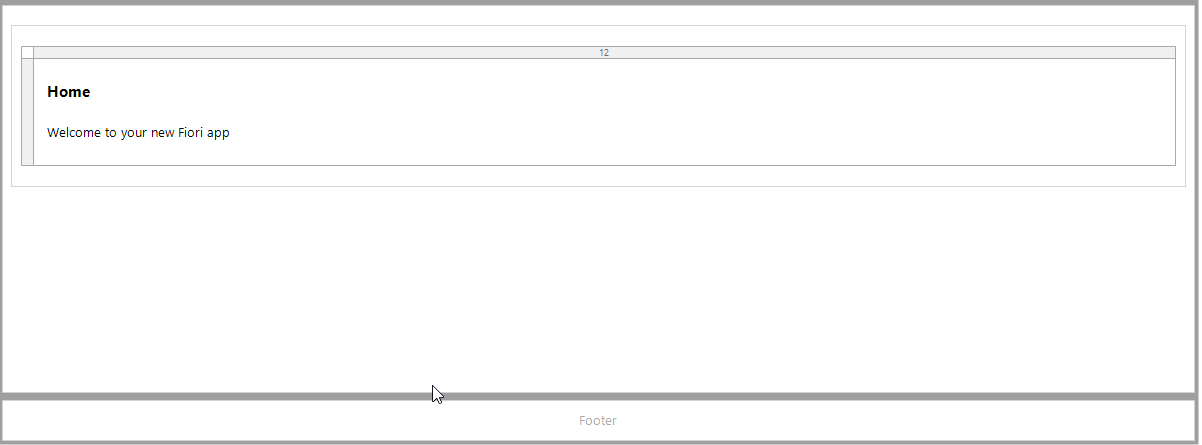


## Configure the Home Page

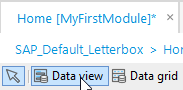
Open the Home page and delete the Layout grid in the center of the page.



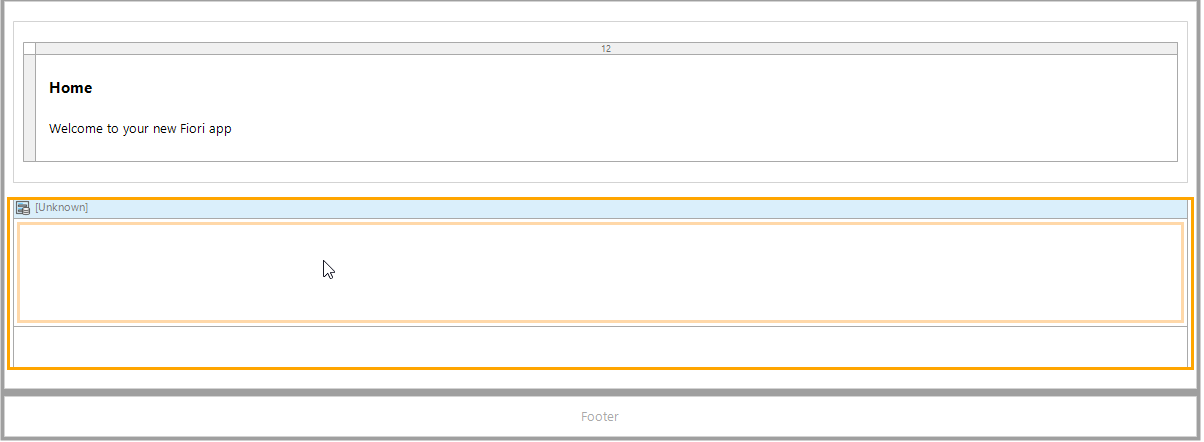
It should look like this:



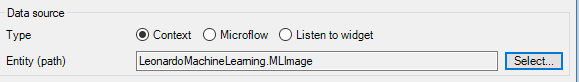
Click the Data view widget above the editor…



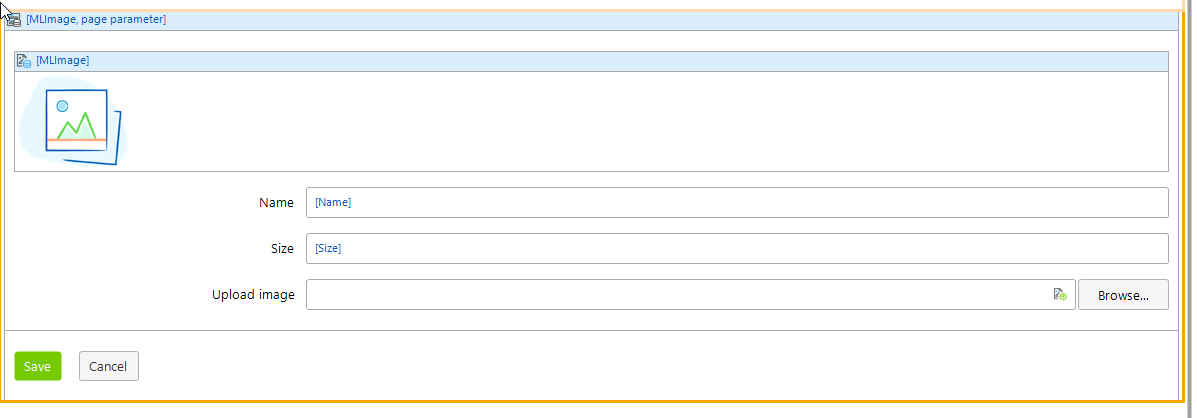
… then click the space where the Layout grid was.



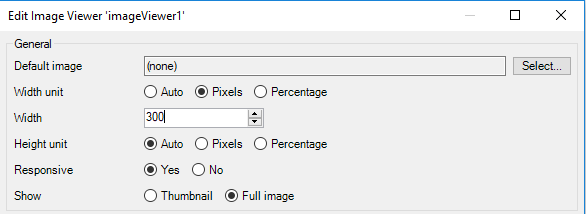
Double-click the header of the data view and configure the Data source as shown.



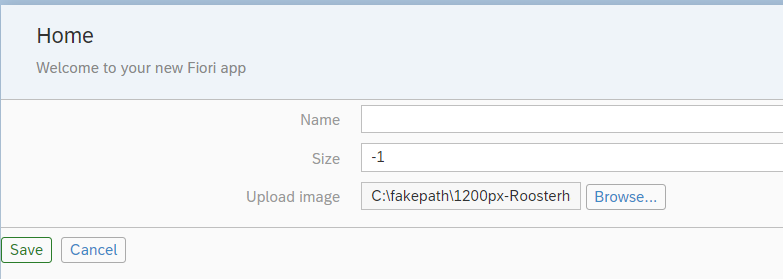
When prompted, select Yes to allow the Mendix Modeler to fill the contents of the Data view.



The widgets added are a Image viewer and Image uploader. Double-click the Image view and configure the image size.



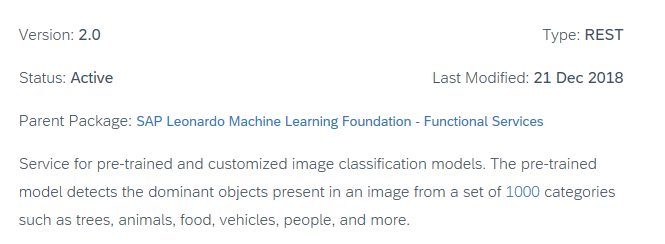
If you run the app now, you can use the Browse button to upload an image but the Image viewer doesn’t automatically show the image.



Next, we’ll create a microflow that will display and classify the image.

## Create the Image Classification Microflow

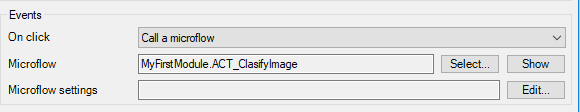
The SAP Leonardo image classification demo service has been pretrained to classify objects in a set of a 1000 categories.



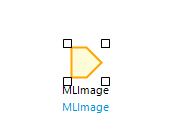
You can find the documentation here:

<https://help.sap.com/viewer/b04a8fe9c04745b98ad8652ccd5d636f/1.0/en-US/d7b4564d6625412c9e50ad1514308043.html>

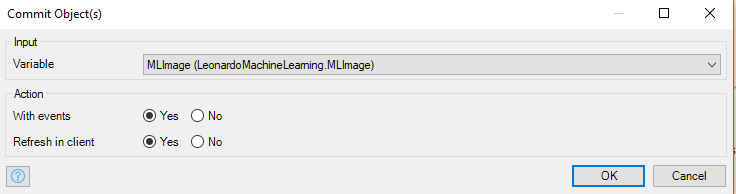
On the Home page, double-click the Save button. Change the caption to Classify Image and configure the Events section as shown below. When you click Select on the Microflow, you will be able to select the Microflows folder and click New to create the microflow. Name it ACT\_ClassifyImage. If you click the Show button, the Modeler will open the microflow for you.



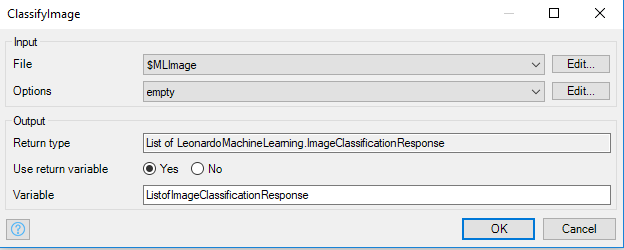
In addition to the usual Start and End events, the microflow includes a Parameter of type MLImage.



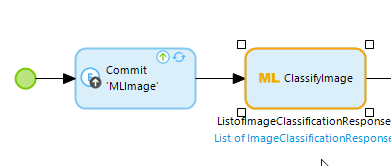
This parameter will be equal to the image object bound to the Image uploader. To make sure the image is displayed, we have to commit it to the database. Add an activity and configure it as a Commit type. Configure it as shown.

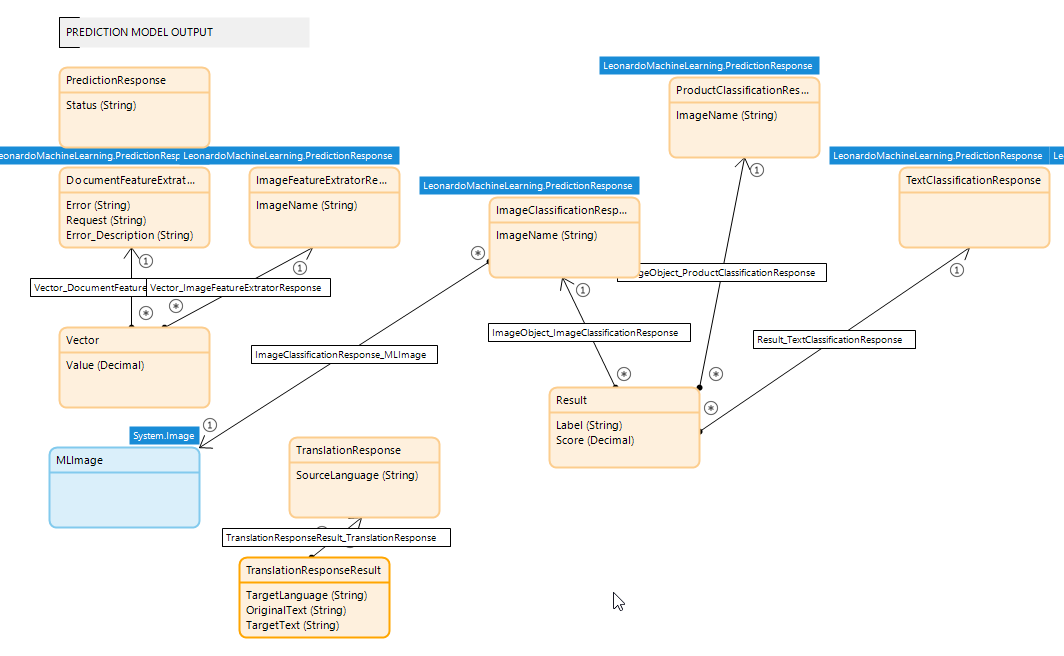


Next, add an activity and set it to the Classify Image type. Configure it as shown:



The microflow so far:



This is sufficient to invoke the image classification service. However, the results are stored in entities included in the Leonardo Machine Learning Domain model. 

Not only is this Domain model complex, it also consists of primarily nonpersistable entities which cannot be bound directly to widgets in the interface. As a results, we have to retrieve the results from these entities and put them in an entity that we can use directly.

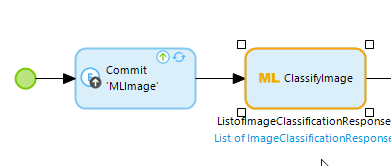
### Create the ImageClassificationResults Entity

Open the MyFirstModule Domain model and add an entity. Configure it as shown in the image below.

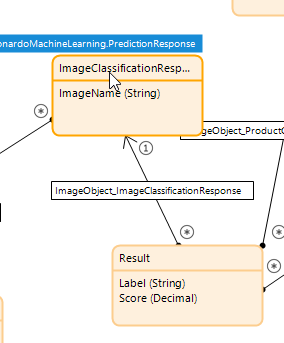


### Complete the Microflow

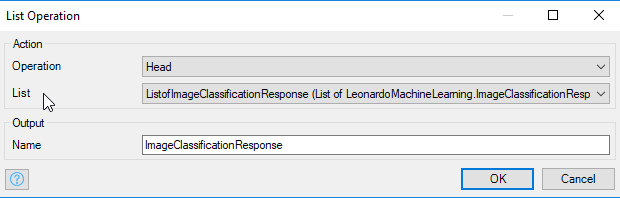
Return to the ACT\_ClassifyImage microflow.



Unfortunately, the ListofImageClassificationResponse variable does not contain the results. The actual results are stored in an entity called Results which is associated with ImageClassificationReponse.

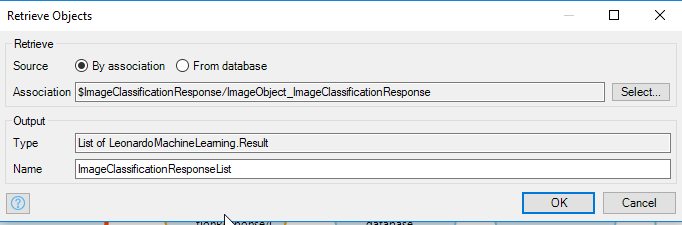


To get the results, insert an activity and make it a List operation. Configure it as shown:

’

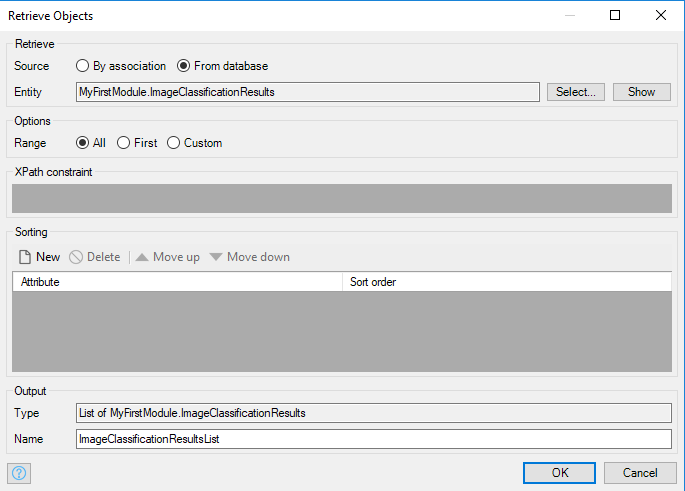
The List is the ListofImageClassificationResponse. This activity will retrieve the first item in the ListofImageClassificationResponse list which will be the record associated with the Results we want to retrieve.

Next, add a Retrieve activity configured as shown:

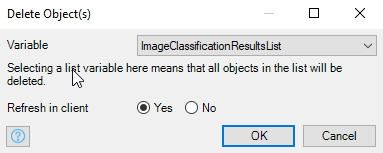


This uses the association between ImageClassificationResponse and Results to retrieve the results.

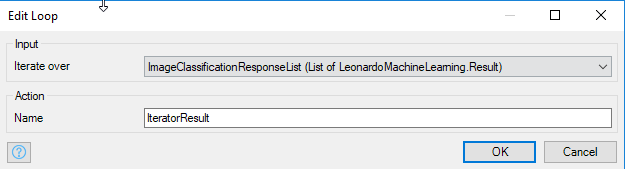
Now, we must add the results to the ImageClassificationResults entity. First, we’ll delete any records that already exist in the entity. Add a Retrieve entity and configure it as shown below. This retrieves all the records in the ImageClassificationResults entity.



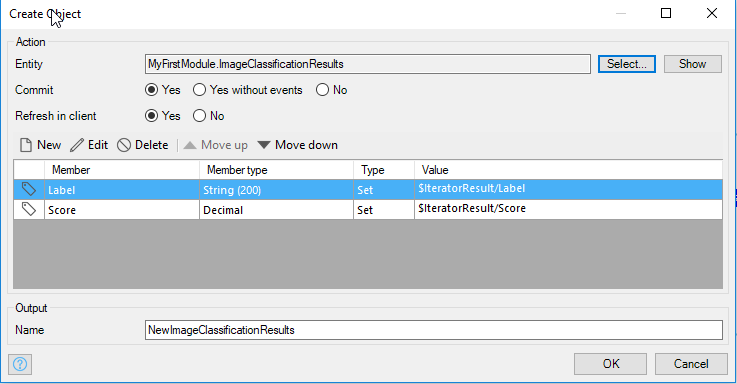
Add a Delete activity configured as follows. This deletes any existing records.



To add the current results to the entity, we must loop through the list returned by the service and create ImageClassificationResults objects. Insert a loop to the right of the Delete activity. Double-click the Iterator and configure it to loop over the ImageClassificationResponseList.

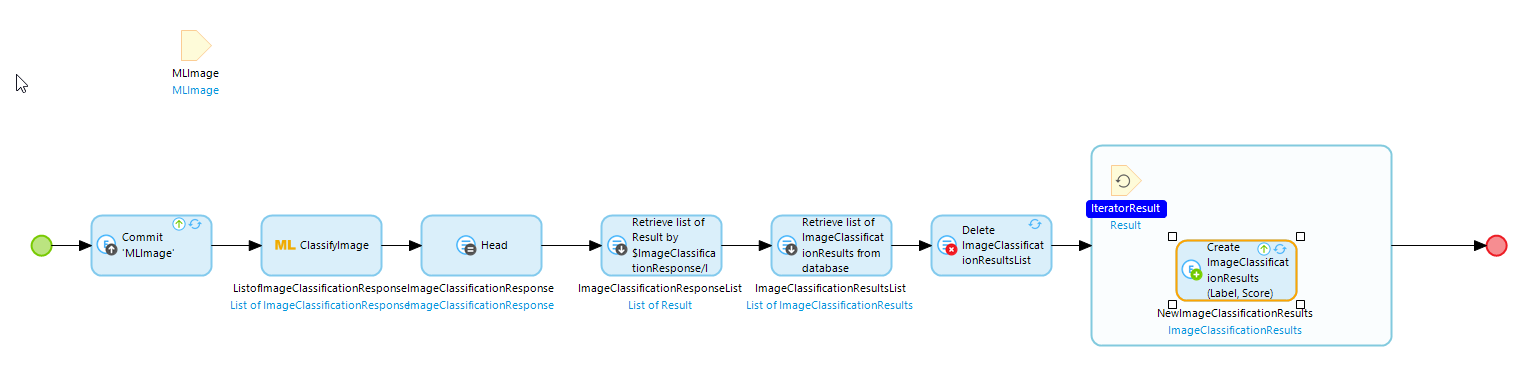


Now insert an activity inside the loop and configure it as a Create object activity.



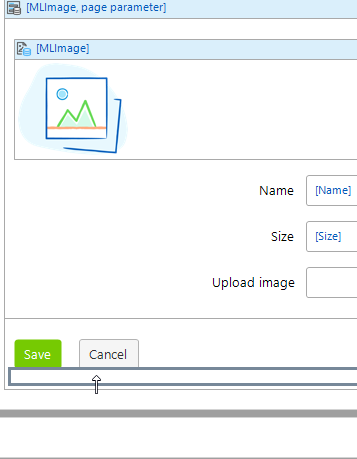
Remember to select Commit (to save it to the database) and Refresh in client (to update the interface). This activity creates a ImageClassificationResults object and saves it to the ImageClassificationResults entity.

The final microflow looks like this:

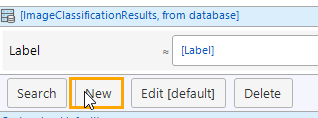


## Add a Data Grid to Display the Results

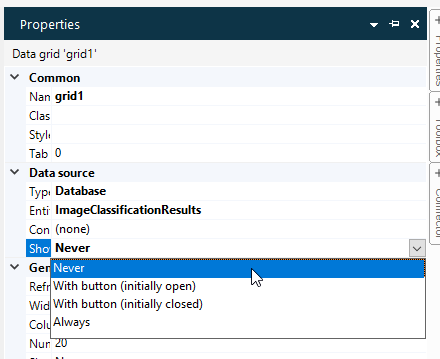
Finally, add a Data grid inside the Data view on the Home page by clicking the Data grid above the editor and the clicking below the Save and Delete buttons as shown.



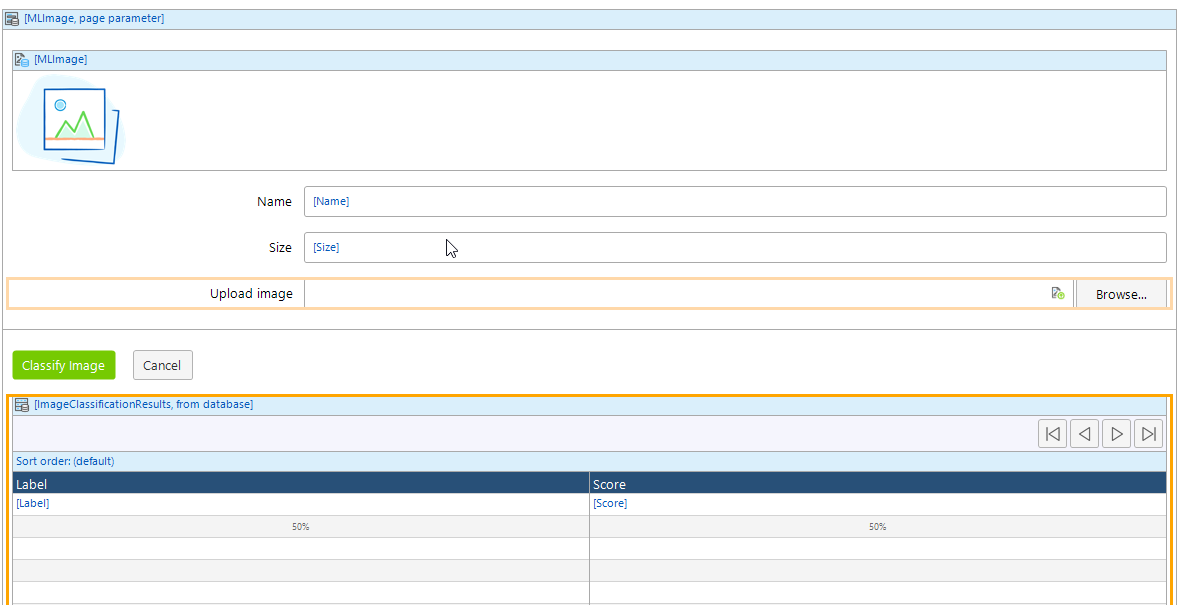
When the Modeler asks if you want to fill the Data grid automatically, choose Yes. We won’t need the buttons so you can delete all of the buttons.



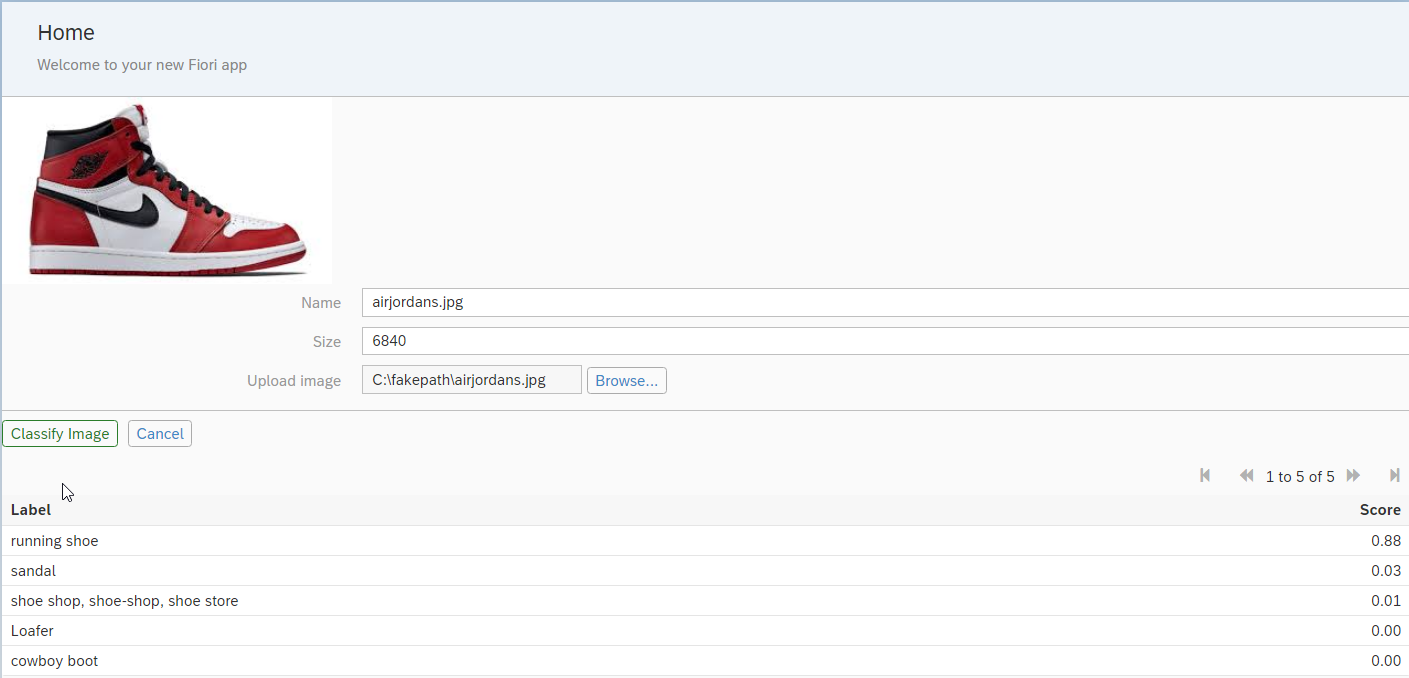
After deleting the buttons, click the header of the Data grid then open the Properties view on the right side of the editor. Set the Show search bar property to Never.



The page should look like this.

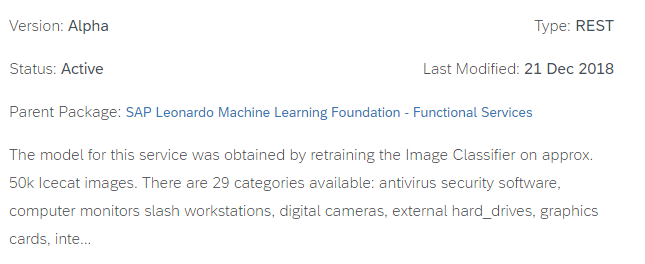


Run the app and you should be able to classify an image.



## Add Product Classification

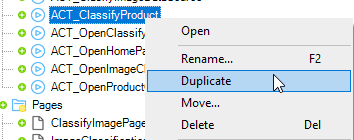
The SAP Leonardo production classification demo service uses the image classification service but has been trained to identify products.



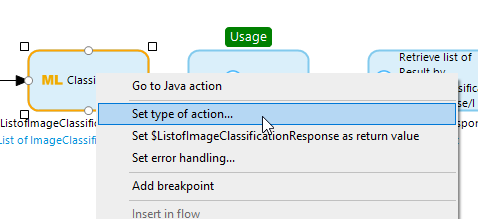
You can find the documentation here:

<https://help.sap.com/viewer/b04a8fe9c04745b98ad8652ccd5d636f/1.0/en-US/3013afaa529440429a6e63dfd31d1799.html>

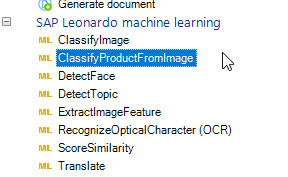
The process for using the product classification service is like image classification. Right-click the ACT\_ClassifyImage microflow and choose Duplicate.



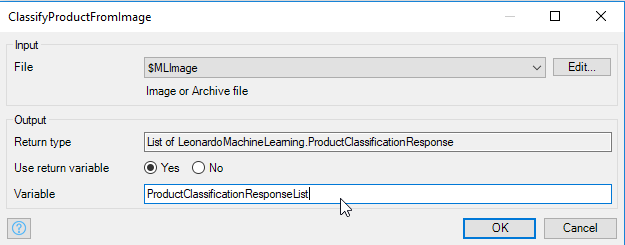
Right-click the microflow and choose Rename to rename the microflow ACT\_ClassifyProduct. Open the new microflow and right-click the ClassifyImage activity and choose Set type of action.



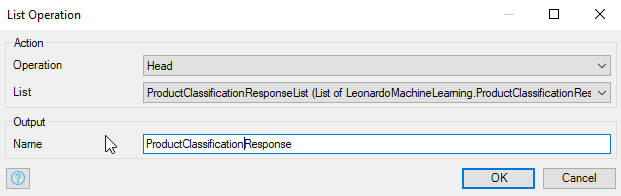
Select the ClassifyProductFromImage type.



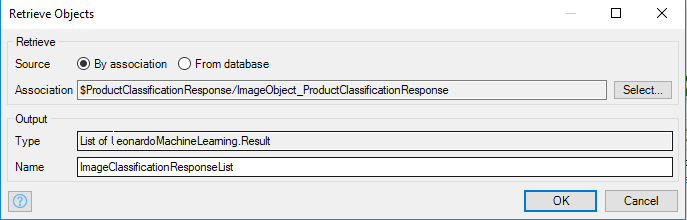
Configure the activity the same as the ClassifyImage activity.



Configure the List operation activity as shown:



Configure the first Retrieve activity as shown:

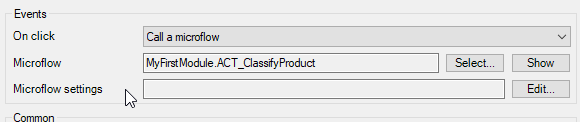


### Update the Home Page

Return to the Home page and double-click the Delete button.



Set the Caption to Classify Product and configure the Event section as shown.



Now you can classify products.

