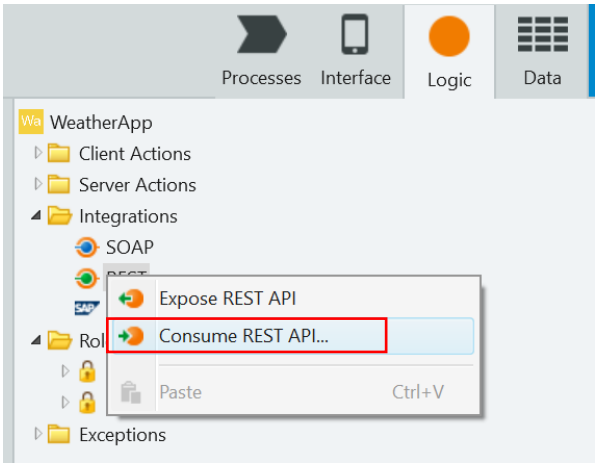
Lab 5: Weather Web Service Mobile Application Lab Guide

Exercise 1: Setting up the Web Service

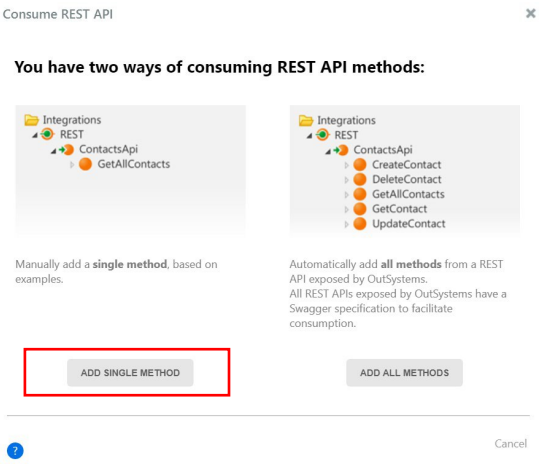
In this exercise, you will create a new application that will retrieve weather information from a JSON web service.

1. Create a new mobile application named **Weather App**, with a **Phone App** Module
    a. In the 'Applications in Development' area, click **New Application**.
    b. In the **New Application** dialog, select **Start from scratch**, and then click **Next**.
    c. Select the **Phone App** template, and then click Next. The templates provide a starting point for the application, containing the layout structure for the mobile app.
    d. Set the Application Name to ***Weather App***.
    e. Type in a simple description for the application.
    f. Select **Create App**.
    g. In the **Modules** area, the list of modules of the application can be found. Specify the **Module Name** as **WeatherApp** and select the **Phone App** module type. Click **Create Module** to create the module.

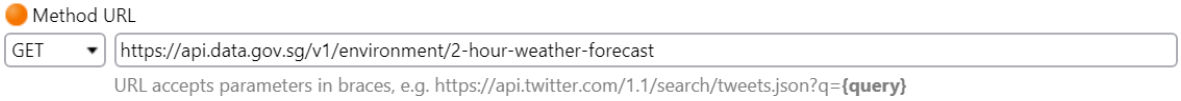
2. Click on the Logic tab, right-click on **REST** and select **Consume REST API…**



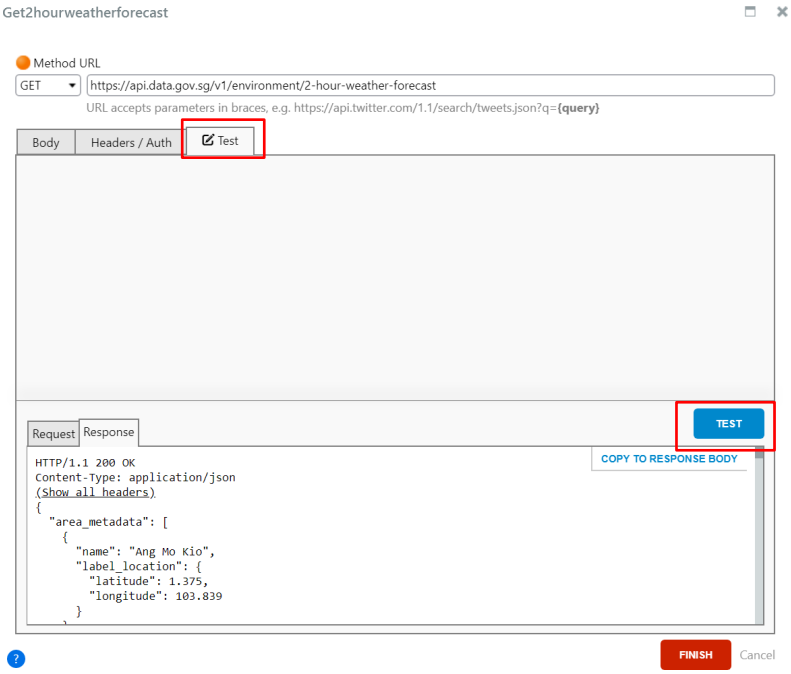
3. Select **ADD SINGLE METHOD**.



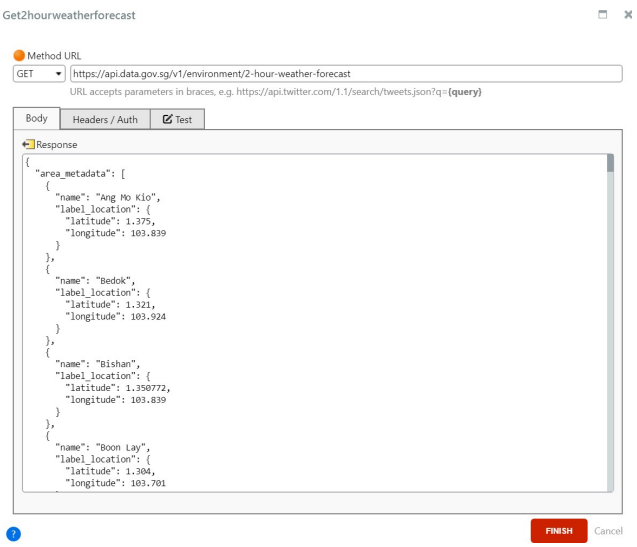
4. In the pop-up dialog, enter the following information:
    a. Method URL: https://api.data.gov.sg/v1/environment/2-hour-weather-forecast



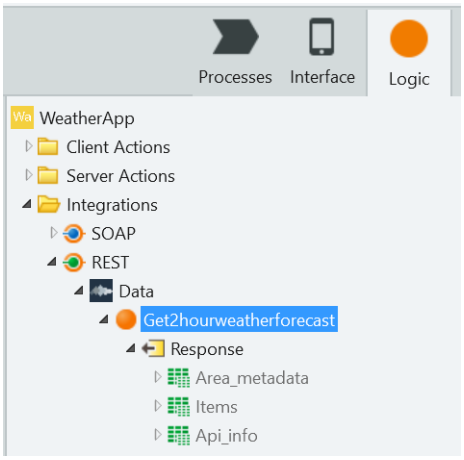
    b. Select the **Test** tab, and click on the **TEST** button. You will see some information populated in the **Response** tab of the bottom panel.



    c. Click on the **COPY TO RESPONSE BODY** button. The result of the web service call will be copied to the Body tab as shown. This information will be used to define the structure of the response of the webservice.



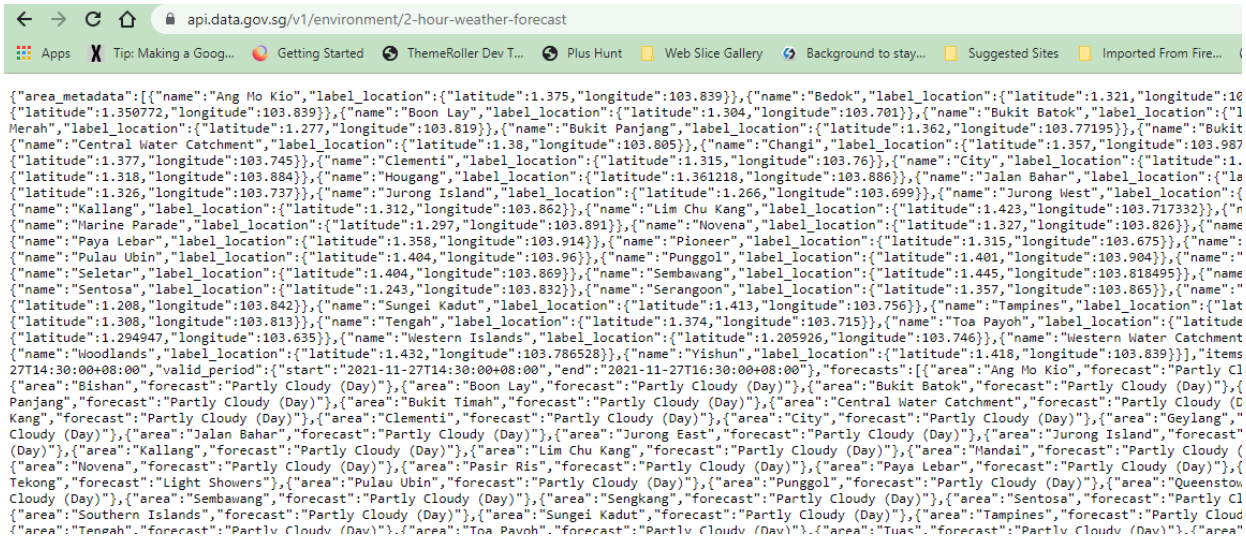
d. Click on the **FINISH** button, the Web Service Action will be created, and you will be able to see them under the REST section.



Exercise 2: Exploring the Web Service Data

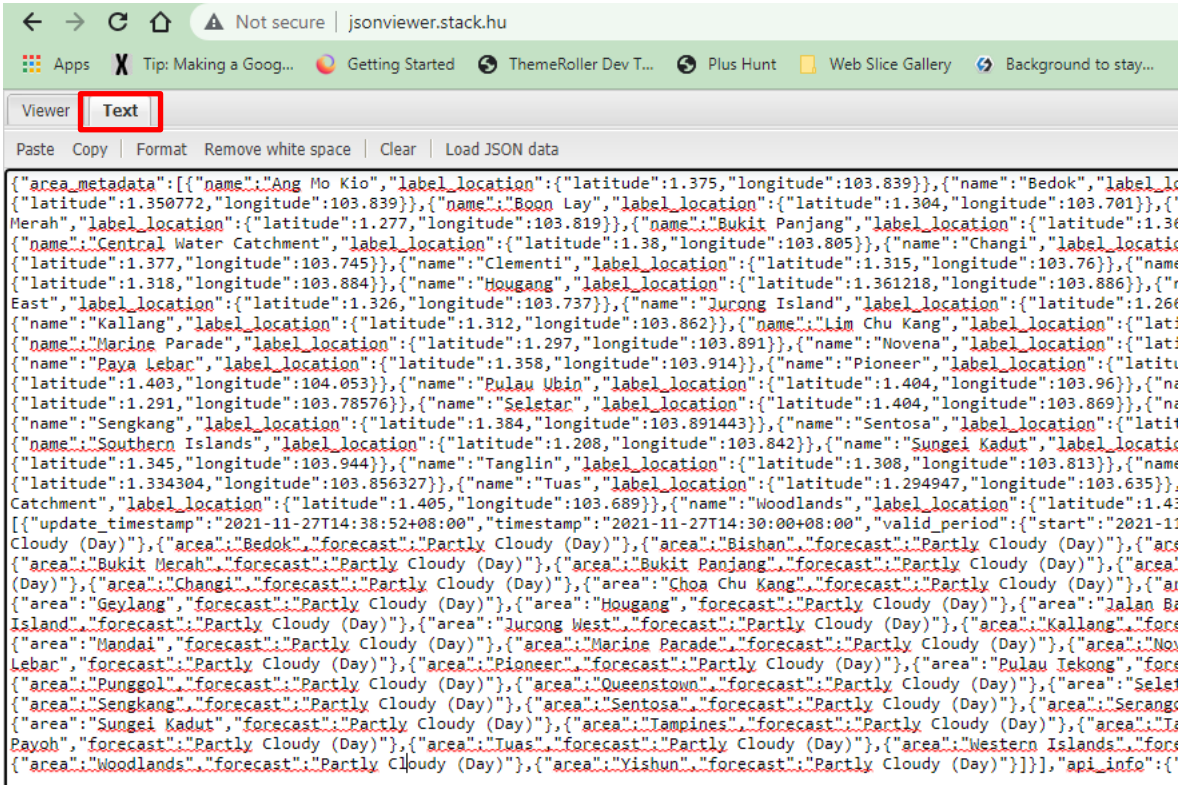
1. Type the following url into the chrome browser:
   a. https://api.data.gov.sg/v1/environment/2-hour-weather-forecast

You will be able to see some text as shown below. The text is in a special format known as JSON.

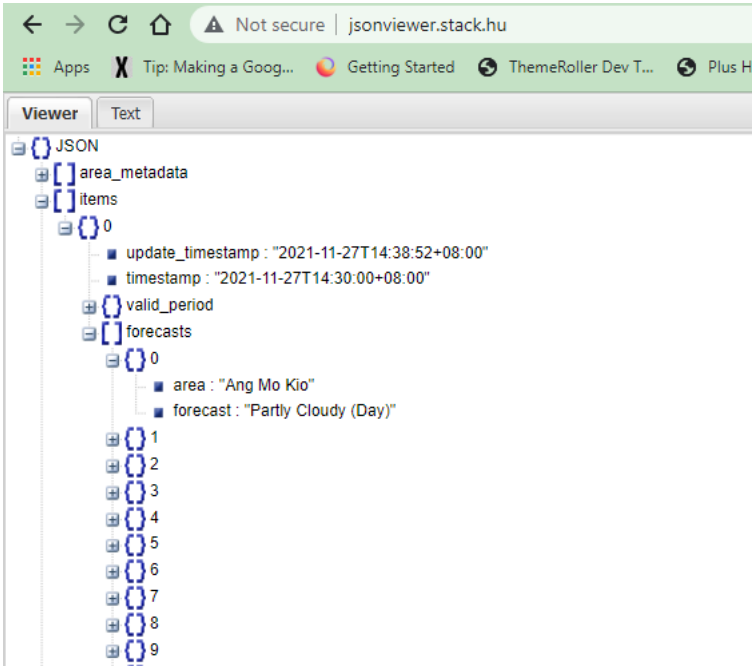


http://jsonviewer.stack.hu/

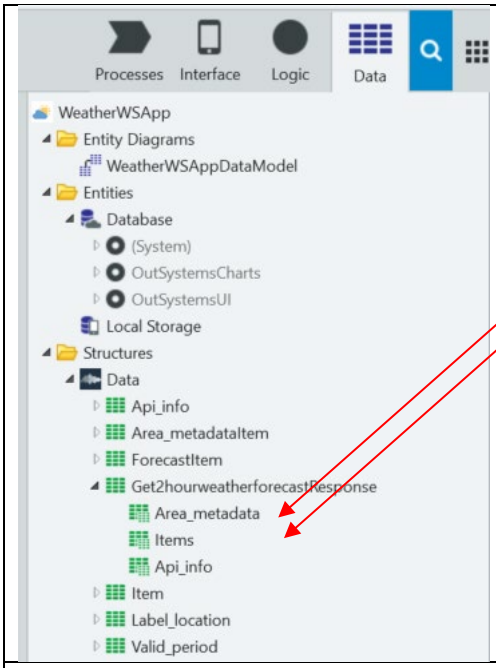
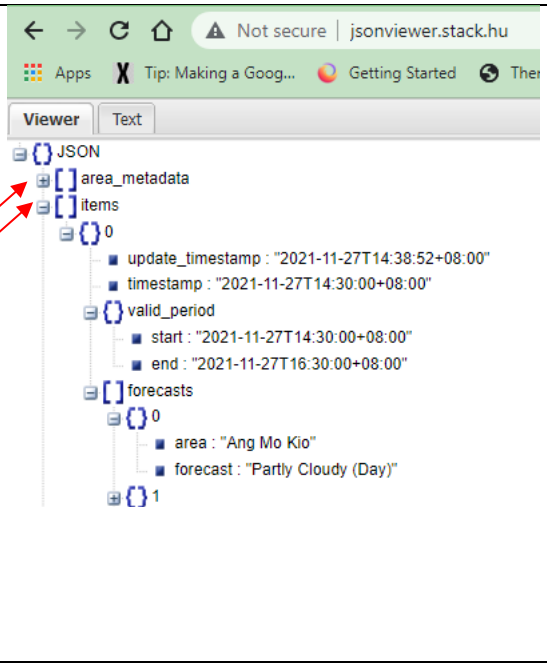
Copy and paste the text from step one into the “Text” tab as shown below:



3. Click on the Viewer tab, you will be able to see the same data presented in a format that is more readable. You may click on the "+" icon or "-" icon to expand or collapse the data. In the screenshot shown below, you will be able to see the weather forecast for "Ang Mo Kio" area. In the next exercise, we will find out how to present the same information in the mobile app.

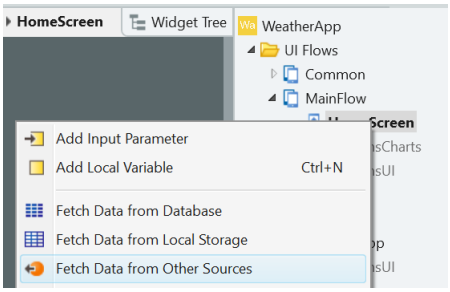


4. Go back to the Outsystems Service studio, click on the Data tab. Compare the data generated by Outsystems when we set-up the weather webservice in Exercise 1 with those that you are in the chrome browser. You will be able to see some similarity between the data.

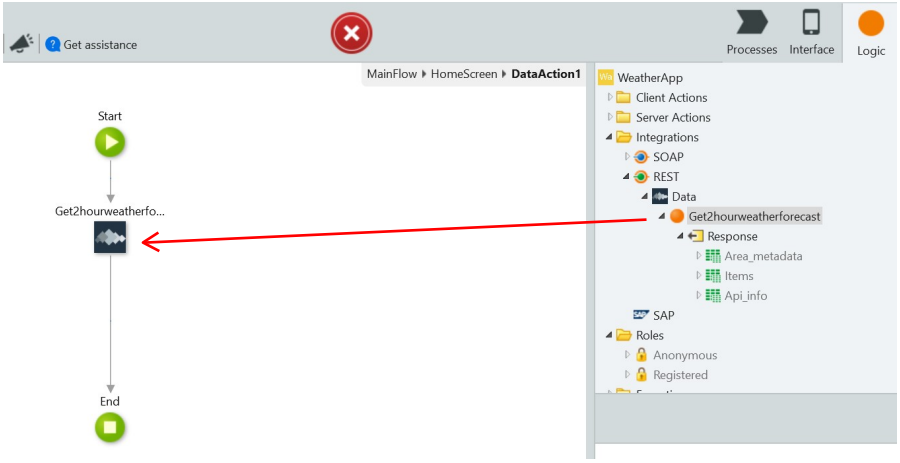
	
Data Structure generated by Outsystems	Data presented in the Chrome browser using JSON viewer

Exercise 3: Showing the Web Service Data

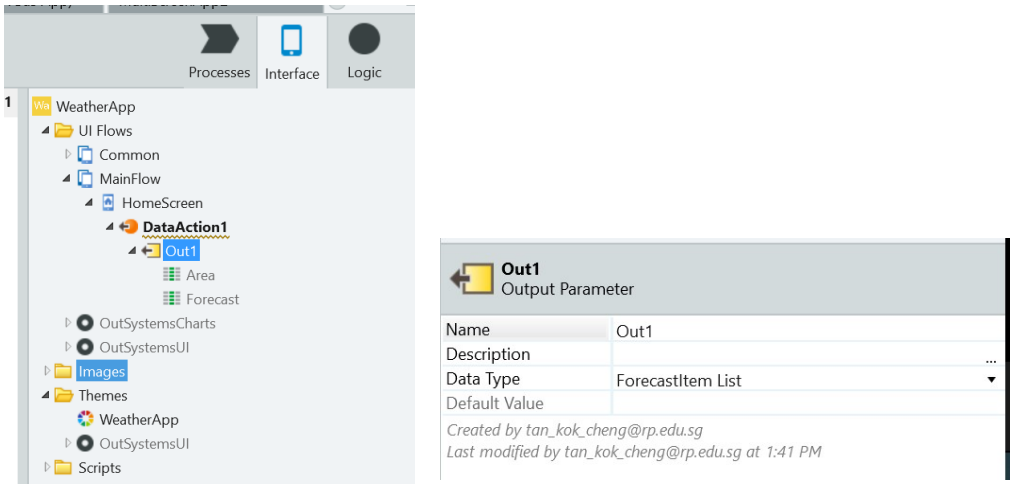
1. Select the **Interface** Tab, create a new screen, and name it as **HomeScreen**.
2. Right-click on **HomeScreen**, and select **Fetch Data from Other Sources**



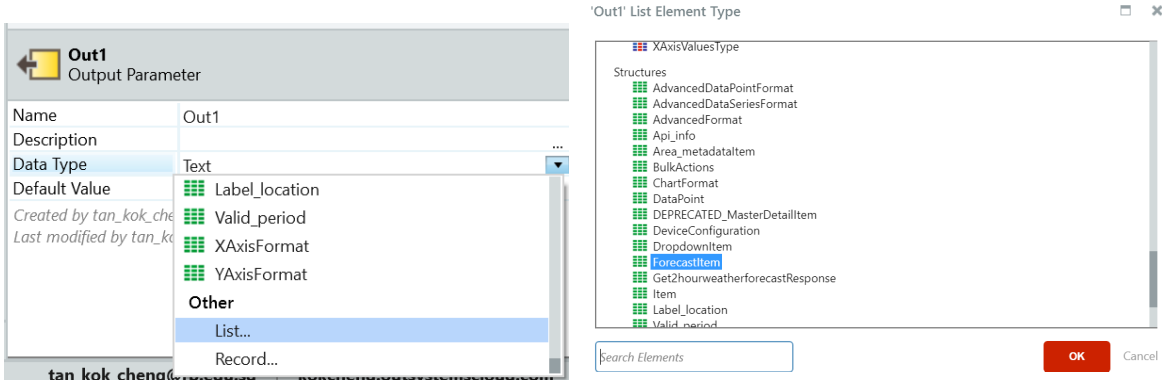
3. Select the Logic Tab, drag and drop the **Get2hourweatherforecast** action to the DataAction1 that you have just created in the previous step when you fetch data from other sources.



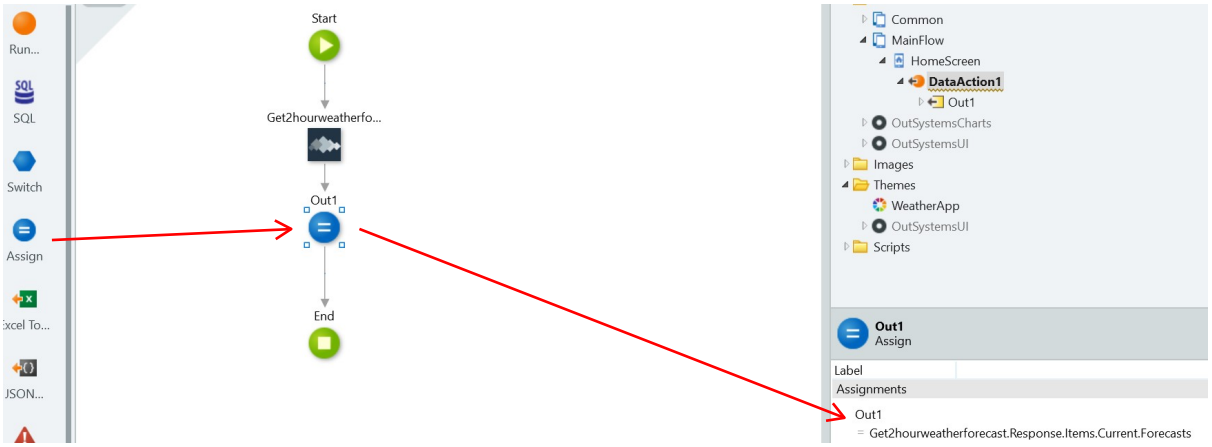
4. Select the **Interface** tab, set the **Out1** Output Parameter to **ForecastItem List**.



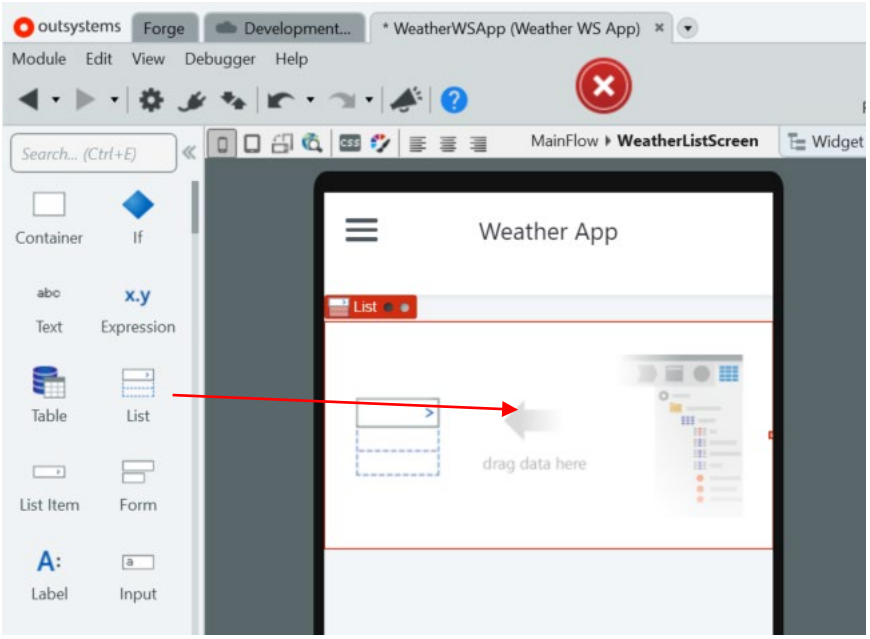
5. To achieve this, click on **Data Type**, scroll down to select **List**. Then, select **ForecastItem** under Structure.



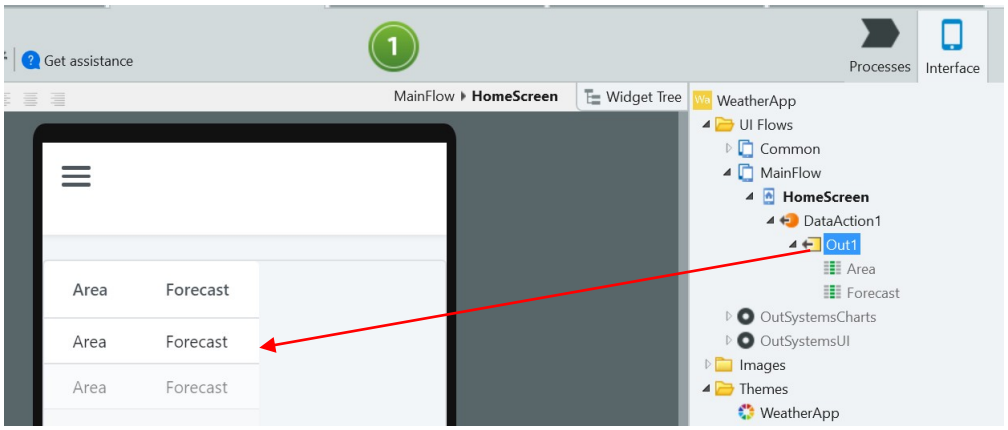
6. Go back to DataAction1, add an Assign widget as shown. Set its value to :
    a. Variable: Out1
    b. Value: Get2hourweatherforecast.Response.Items.Current.Forecasts



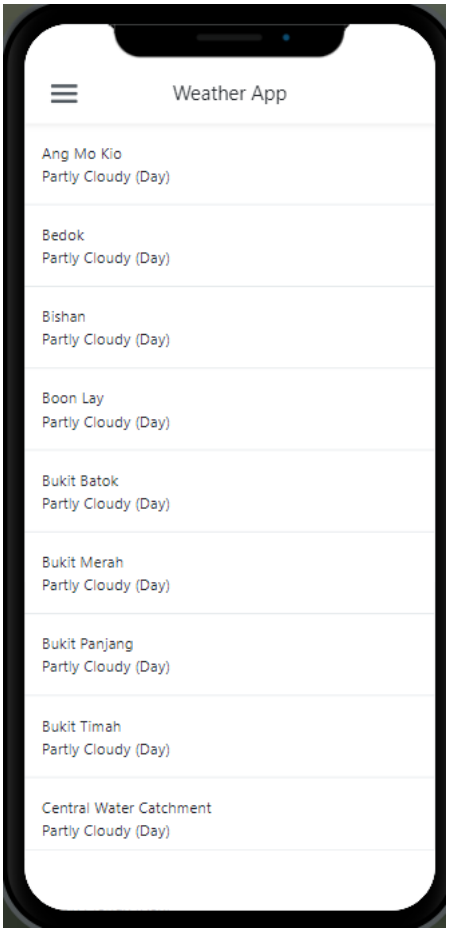
7. Drag a List to the HomeScreen as shown.



8. Select the Interface tab, drag Out1 into the List as shown.

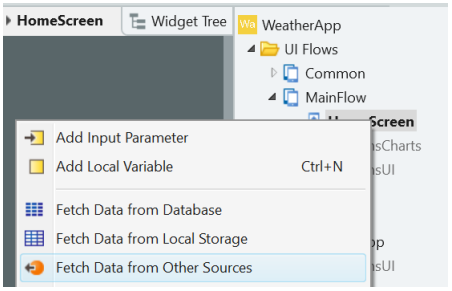


9. You may click on the  1-Click Publish button to update and test the app. You will be able to see a list with the area and weather forecast as shown.

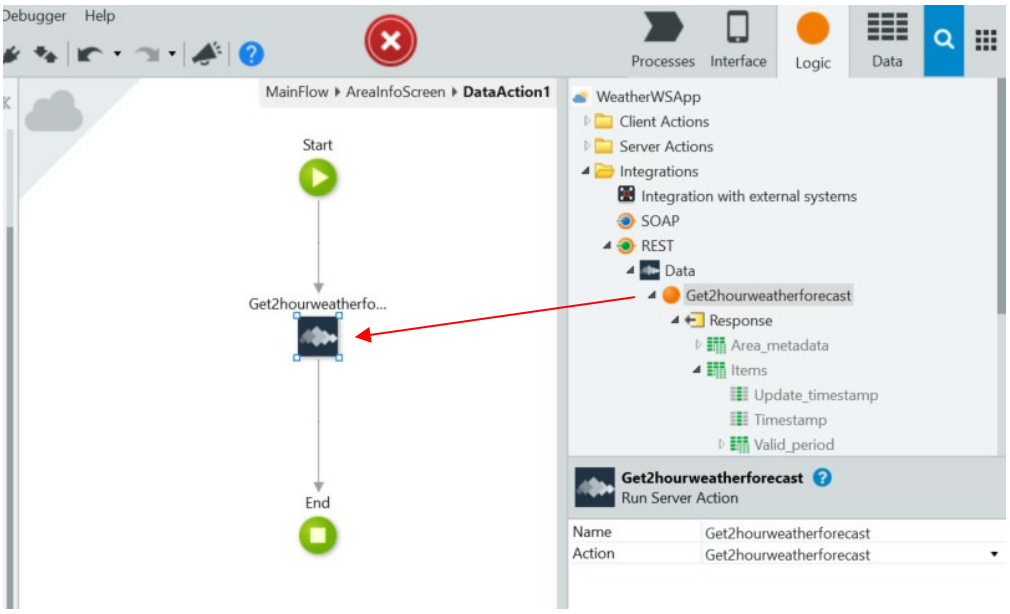


Exercise 4: Extracting Update Time Information from the Weather Web Service

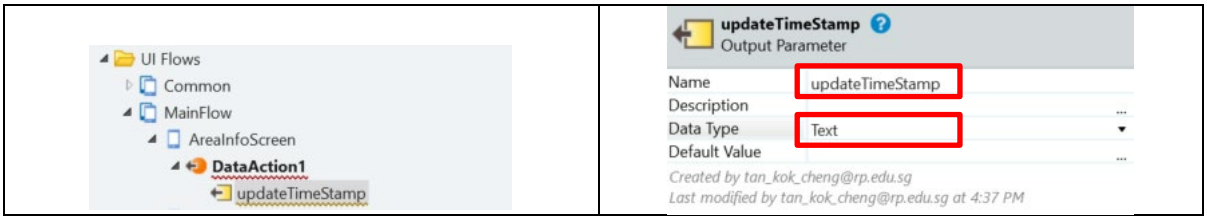
1. Select the **Interface** Tab, create a new screen, and name it as **AreaInfoScreen**.



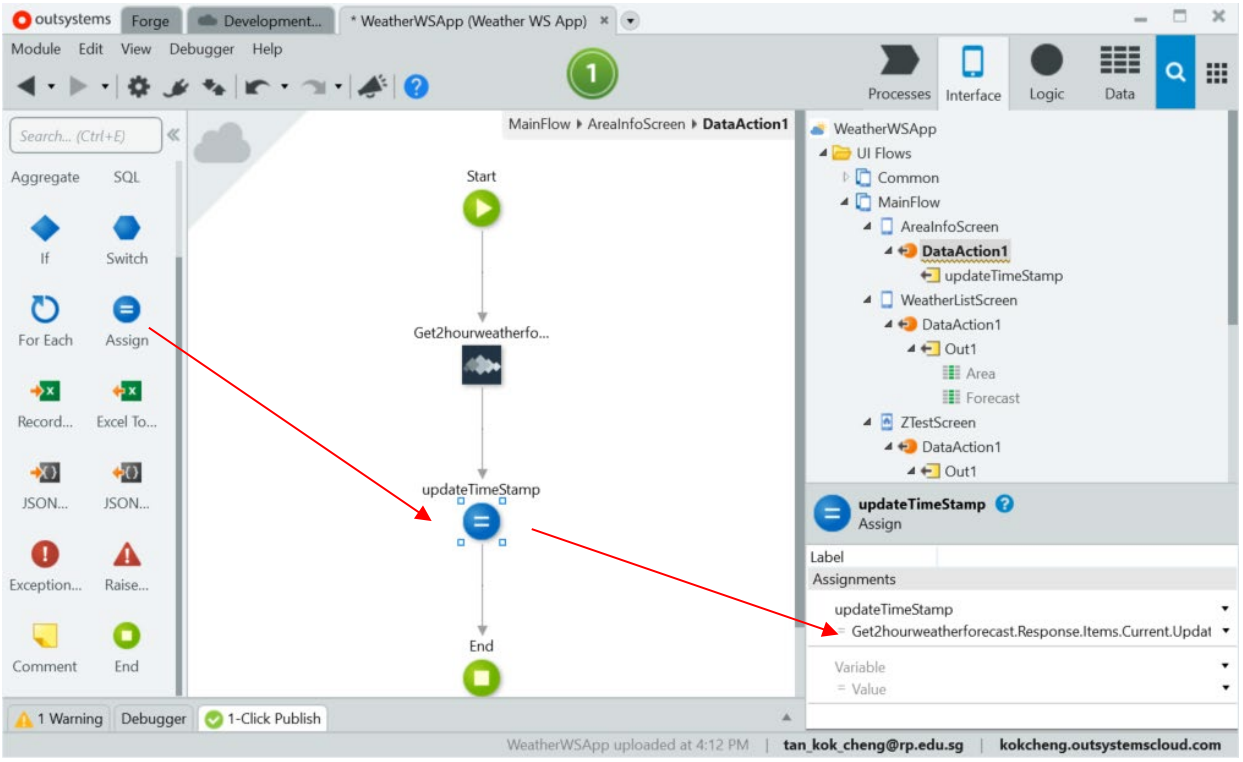
3. Select the Logic Tab, drag and drop the **Get2hourweatherforecast** action to the DataAction1 that you have just created in the previous step when you fetch data from other sources.



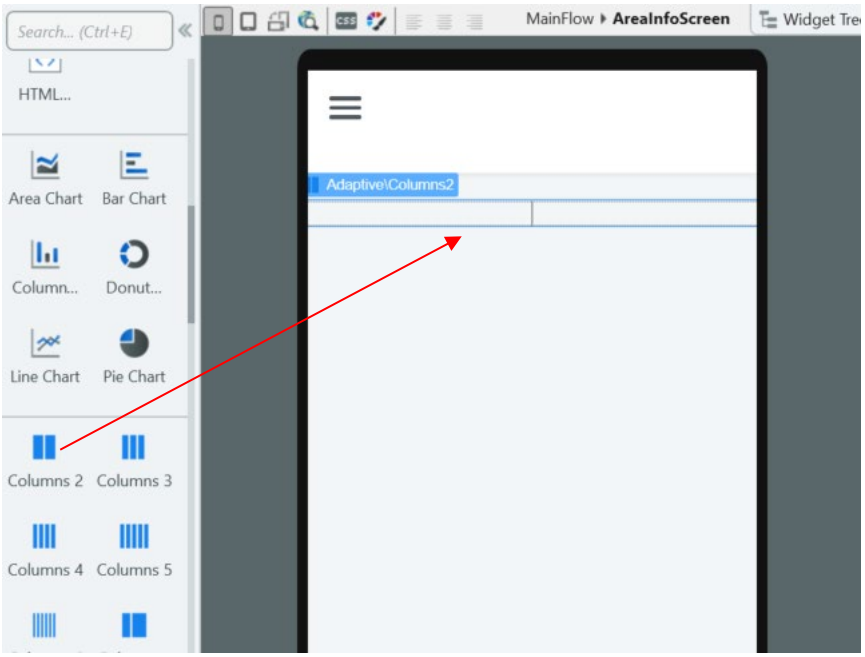
4. Select the **Interface** tab, change the **Out1** Output Parameter to **updateTimeStamp**. The Data Type will be set to **Text**.



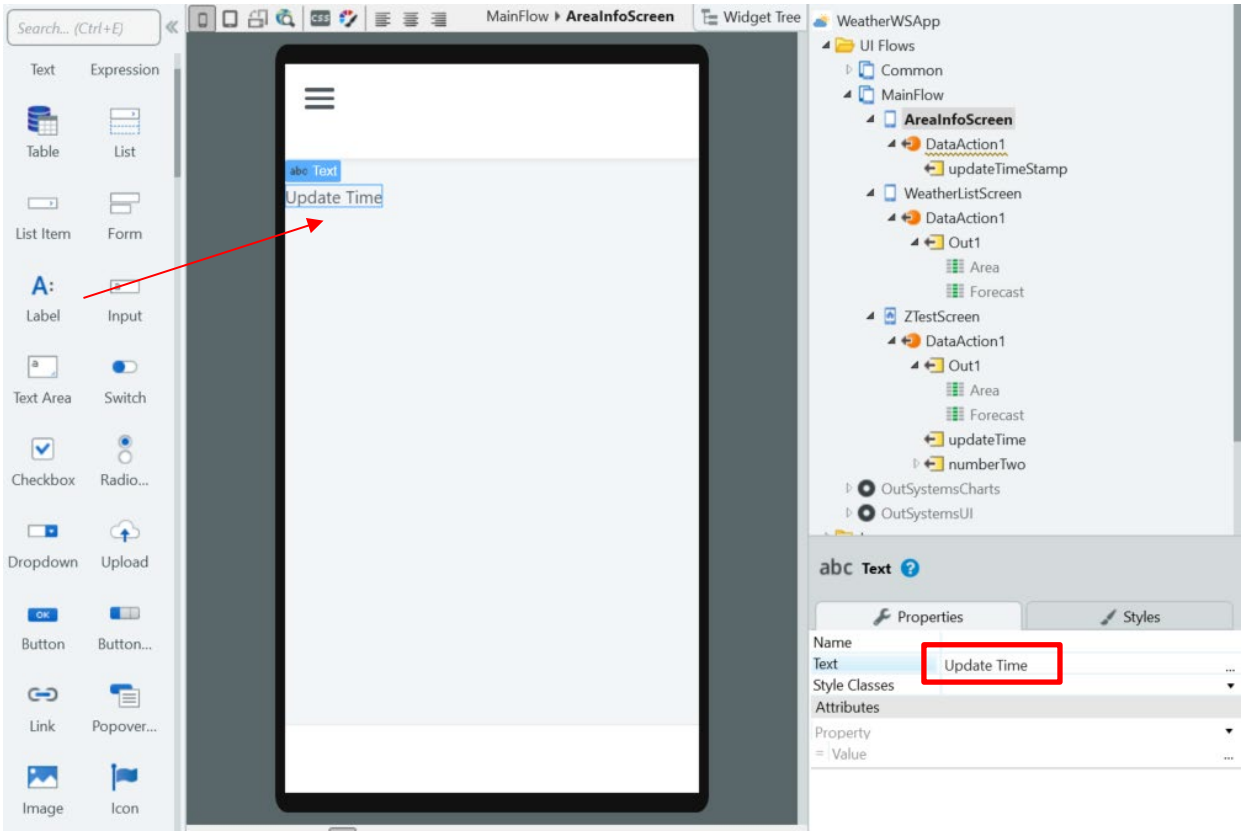
5. Go back to DataAction1, add an Assign widget as shown. Set its value to :
   a. Variable: updateTimeStamp
   b. Value: Get2hourweatherforecast.Response.Items.Current.Update_timestamp



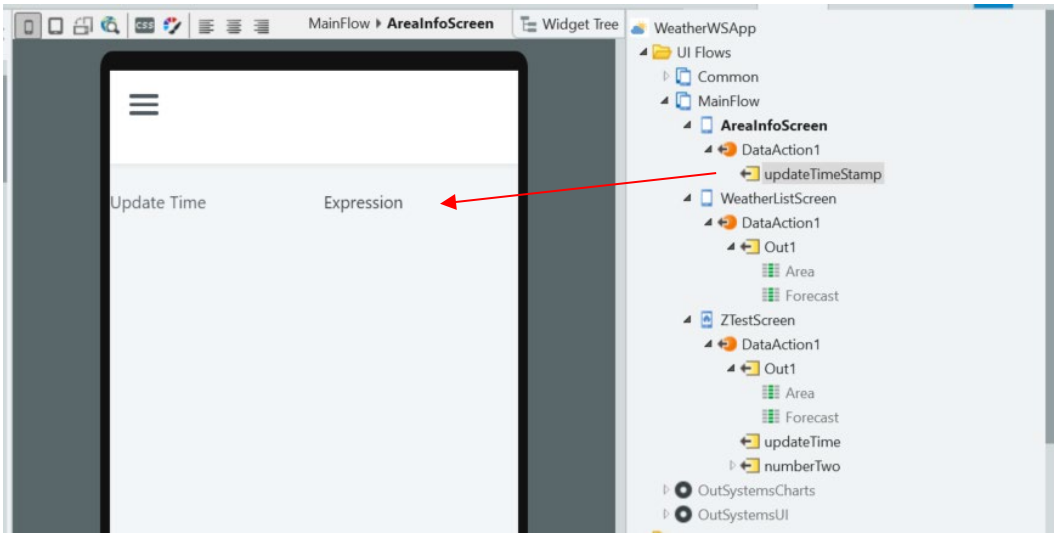
6. Click on the Interface tab, followed by the AreaInfoScreen. Add a new **Columns 2** to the screen.



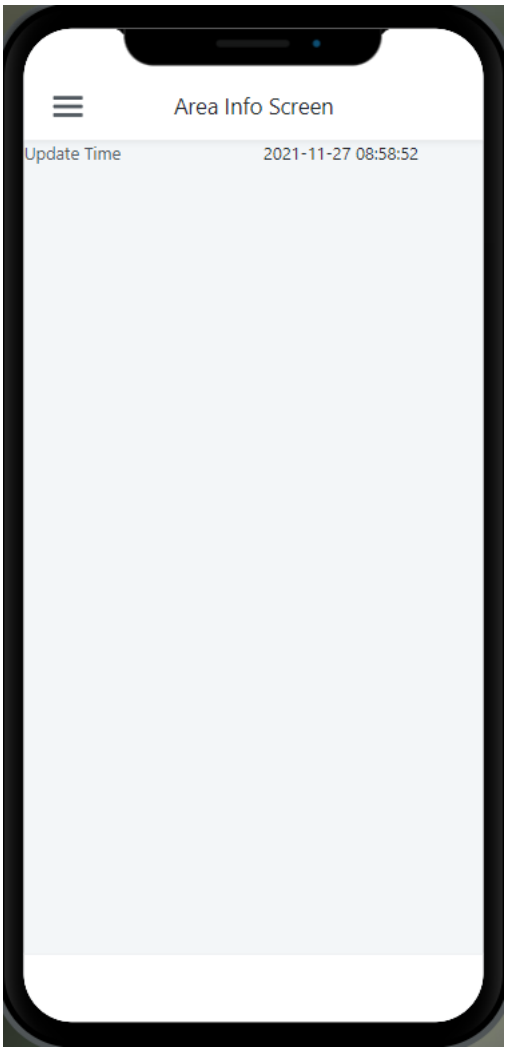
7. Add a Label to the left-hand column, and change the **Text** to "**Update Time**".



8. Drag the updateTimeStamp variable into the right-hand column. You should see an Expression created as shown.



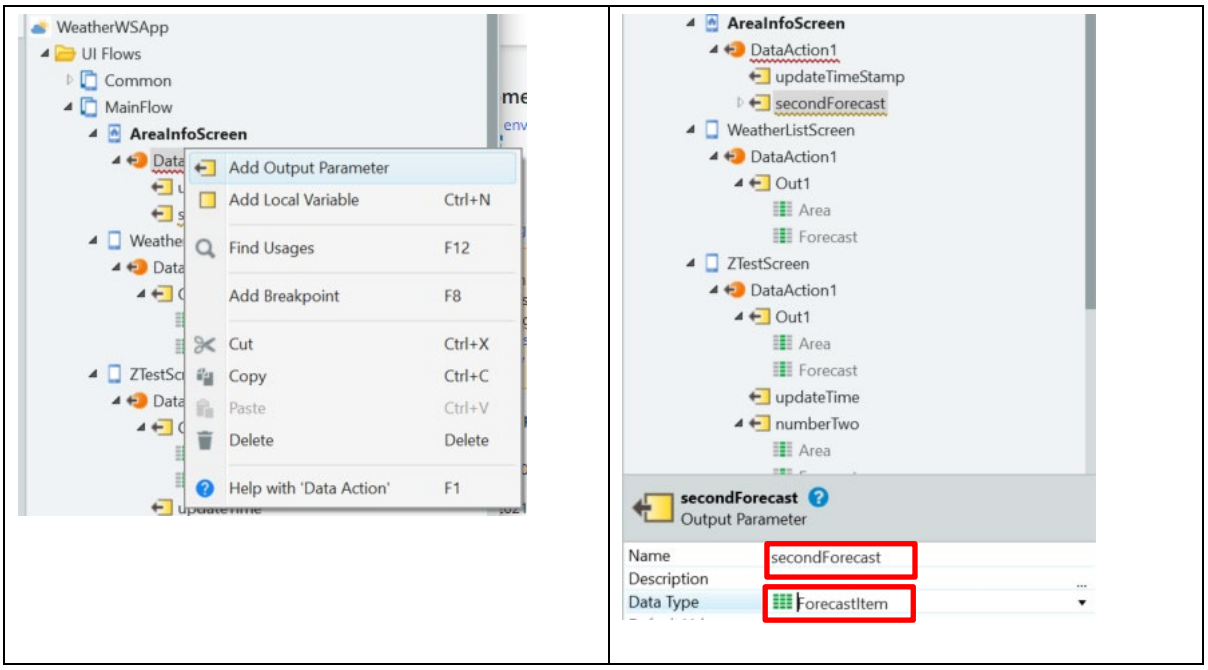
9. You may click on the  1-Click Publish button to update and test the app. You will be able to see the date and time that the weather information is updated as shown.



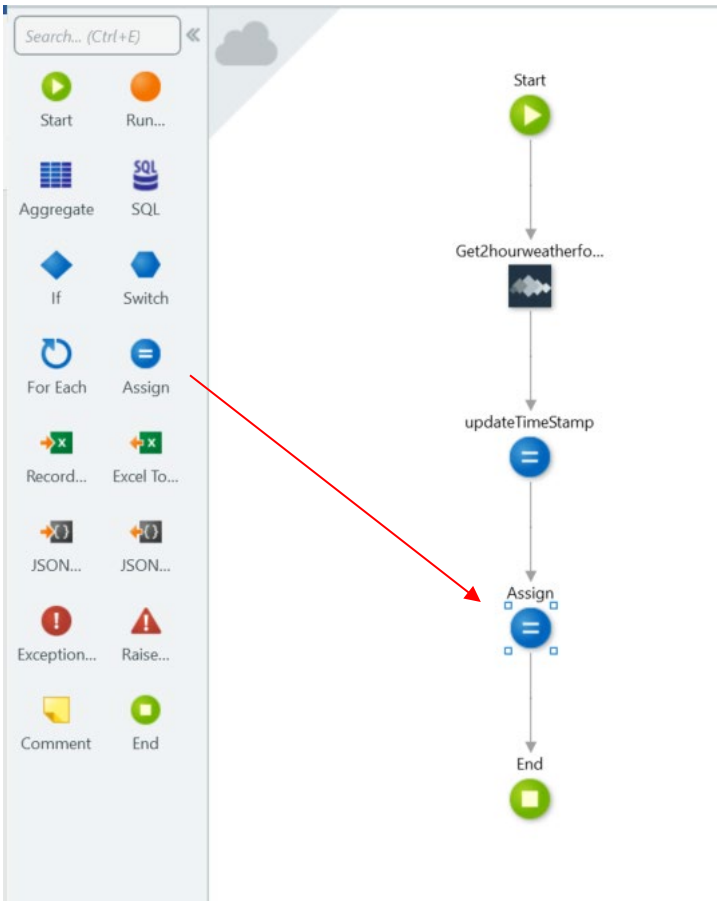
Exercise 5: Extracting Weather Forecast for Specific Area

1. In exercise 3, we displayed the weather information for all the areas in Singapore inside a list. In this exercise, we will learn how to get and display the weather information for specific area.

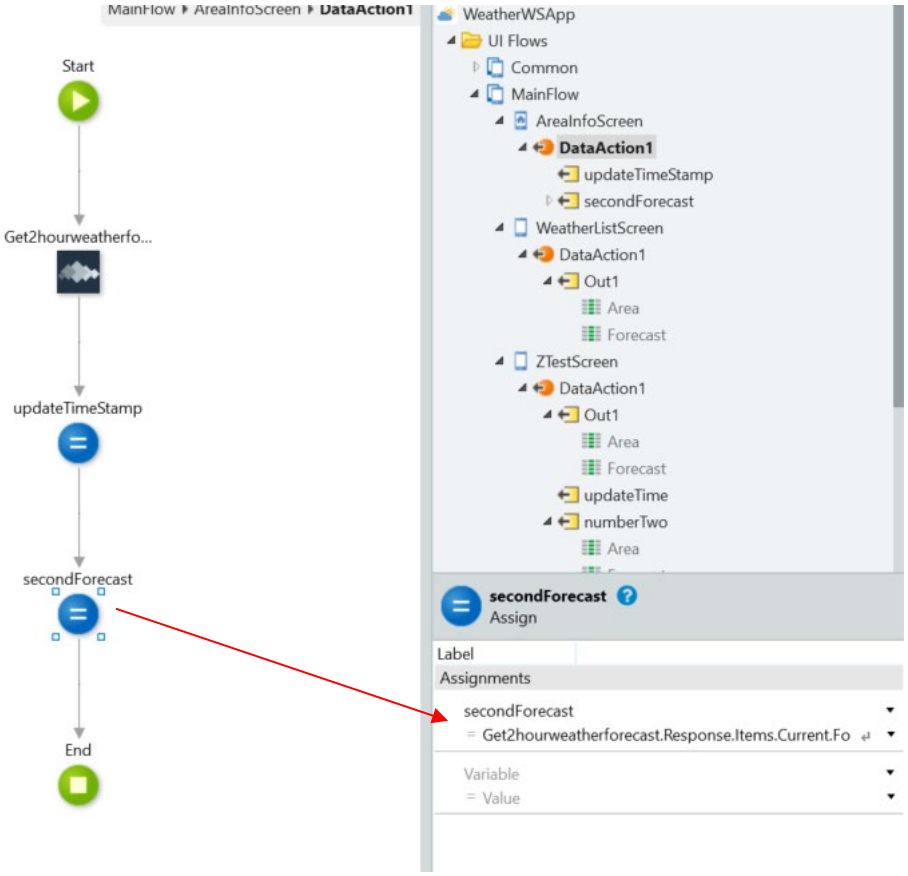
2. Right click on **DataAction1** under **AreaInfoScreen**, select **Add Output Parameter**. Rename this output parameter from **Out1** to **secondForecast**. Change the Data Type to **ForecastItem**.



3. Double-click on **DataAction1** to open the workflow screen as shown. Add another **assign** widget as shown.

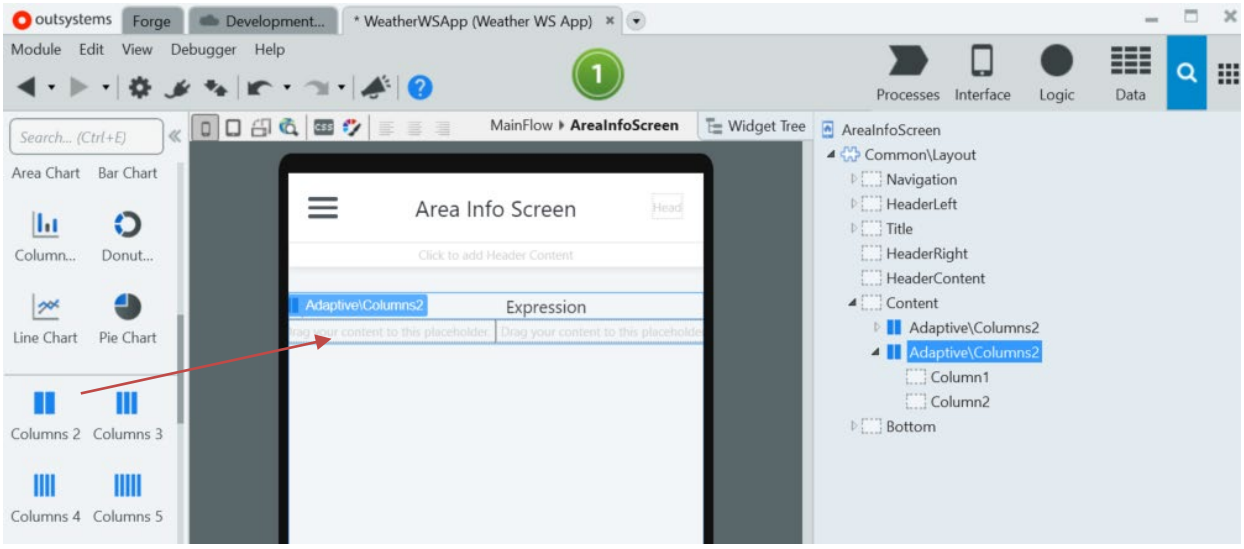


4. Select the **Assign** widget as shown. Set its value to:
    c. Variable: secondForecast
    d. Value: Get2hourweatherforecast.Response.Items.Current.Forecasts[1]

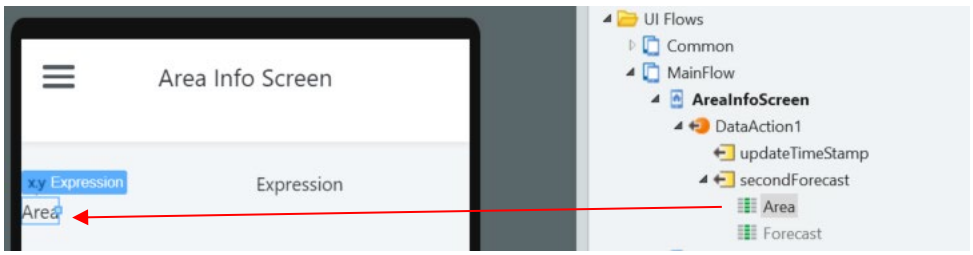


Note: The Forecast[1] means we would like to get the second area in the list. In this case, we will be getting the forecast for Bedok area.

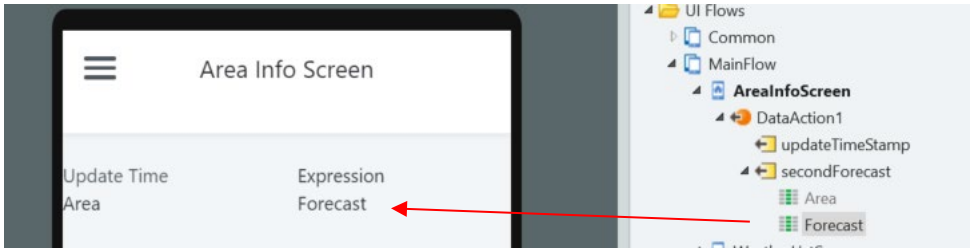
5. Click on the **Interface** tab, select **AreaInfoScreen**, add another **columns 2** widget just below the update time label.



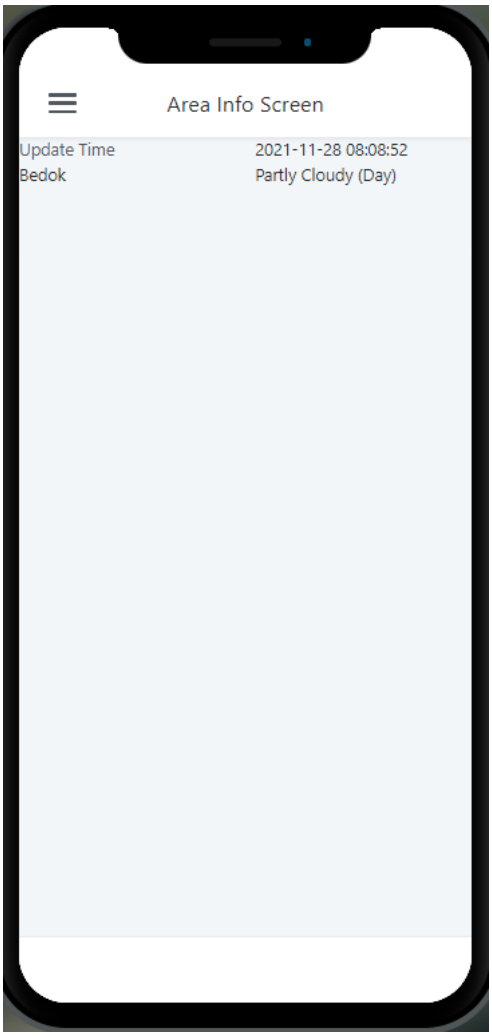
6. Select the Area data under the **secondForecast** variable, drag it to the left-hand column as shown.



7. Repeat the same step for the Forecast variable. Drag it to the right-hand column as shown.



8. You may click on the 1-Click Publish button to update and test the app. You will be able to see the date and time that the weather information is updated as well as the weather forecast for the Bedok area.


Area Info Screen
Update Time
2021-11-28 08:08:52
Bedok
Partly Cloudy (Day)

End of Lab 5