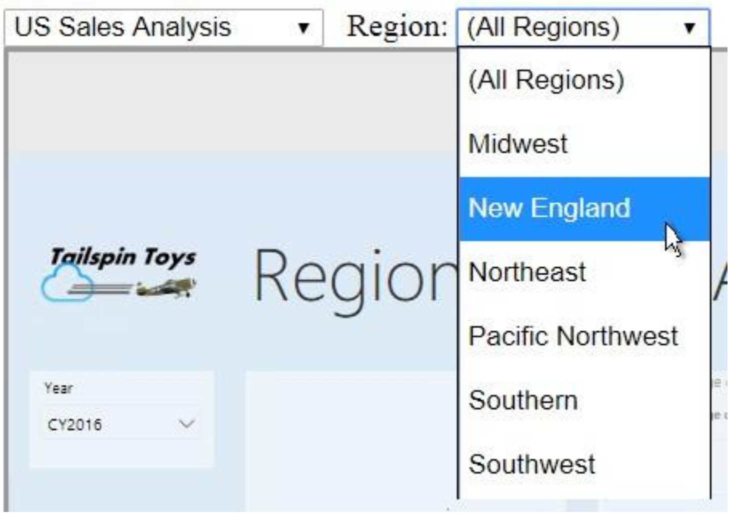
Exploring the Power BI JavaScript API

Overview

The estimated time to complete the lab is 30 minutes

*You must have completed **Lab 02A** before commencing this lab.*

In this lab, you will explore the sample web application and various Power BI JavaScript API operations. You will then add client-side filtering to the **EmbedReport.aspx** web form to enable filtering by region.



Exploring the Sample Web Application

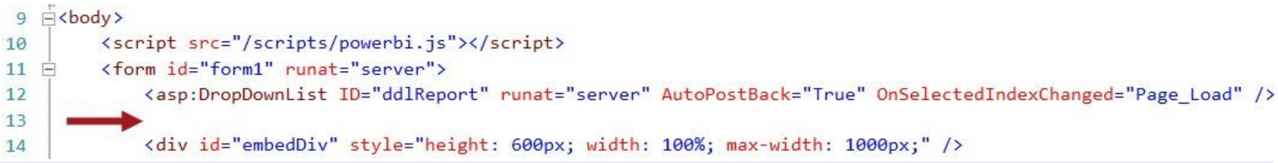
In this exercise, you will explore the **Report Embed Sample** web application to explore several capabilities supported by the JavaScript API.

Outputting Embed Values

In this task, you will modify the **EmbedReport.aspx** web form logic to output specific embed variable values.

1. In Visual Studio, in the **PowerBIEmbedding** project, open the **EmbedReport.aspx** item.
2. To output variable values, beneath the **asp:DropDownLists** element, add the following HTML elements:

   For convenience, the elements can be copied from the ***<CourseFolder>\PowerBIDevIAD\Lab02B\Assets\Snippets.txt*** *file.*



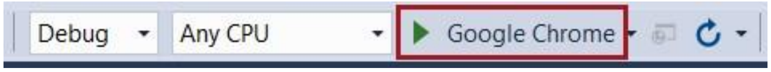
```
<body>
    <script src="/scripts/powerbi.js"></script>
    <form id="form1" runat="server">
        <asp:DropDownList ID="ddlReport" runat="server" AutoPostBack="True" OnSelectedIndexChanged="Page_Load" />

        <div id="embedDiv" style="height: 600px; width: 100%; max-width: 1000px;" />
```

ASP.NET

```
<br />
<span>Embed Token: </span>
<textarea cols="120" rows="1" readonly="readonly"><% = this.embedToken %>
<span>Embed URL: </span>
<textarea cols="120" rows="1" readonly="readonly"><% = this.embedUrl %>
<span>Report Id: </span>
<textarea cols="120" rows="1" readonly="readonly"><% = this.reportId %></textarea><br />
```

3. On the toolbar, start the web application (start debugging).



4. When the web browser opens, navigate to the **Embed Reports** page.

Exploring the Sample Web Application

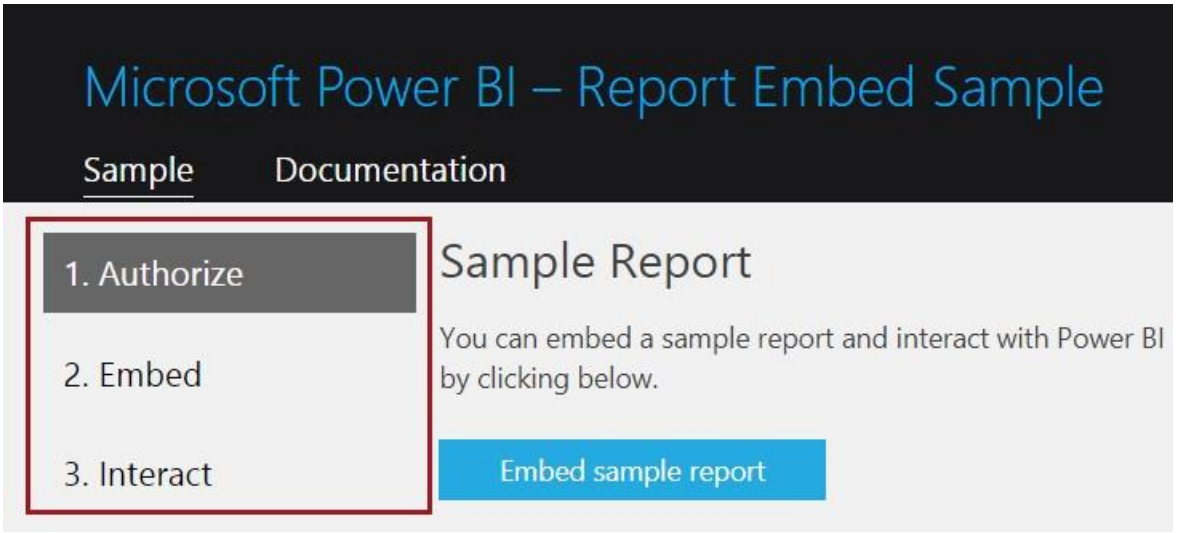
In this task, you will explore the **Report Embed Sample** web application.

1. In a new Google Chrome window, navigate to http://pbiembedsamplev3.azurewebsites.net/v2-demo/index.html.

   For convenience, the URL can be copied from the ***<CourseFolder>\PowerBIDevIAD\Lab02B\Assets\Snippets.txt*** *file.*

   The sample web application was designed to enable experimenting with Power BI embedded analytics without developing an application. It works with either a sample report or dashboard, or with your report or dashboard. It also allows creating and editing reports. In this lab, you will embed Power BI content from your app workspace.

2. In the left pane, notice the workflow that the sample web application supports.

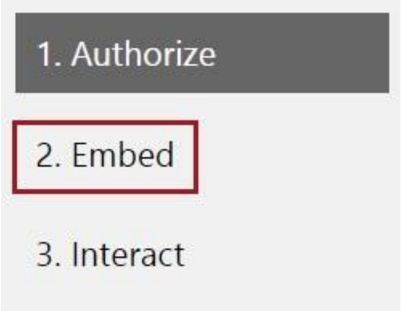


Microsoft Power BI – Report Embed Sample

Sample Report

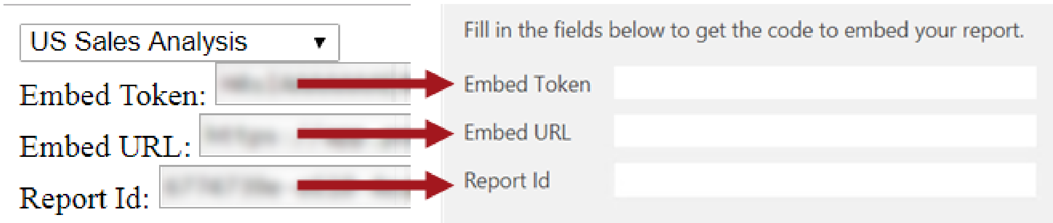
You can embed a sample report and interact with Power BI by clicking below.

3. Select the **Embed** page.



4. Copy all embed values from your web app into the three boxes in the sample web application.

   *When copying the values, be sure to select all text in the boxes (press **Ctrl+A**).*



US Sales Analysis ▾

Embed Token: [redacted]

Embed URL: [redacted]

Report Id: [redacted]

Fill in the fields below to get the code to embed your report.

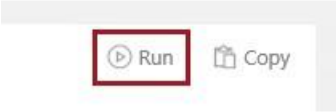
Embed Token

Embed URL

Report Id

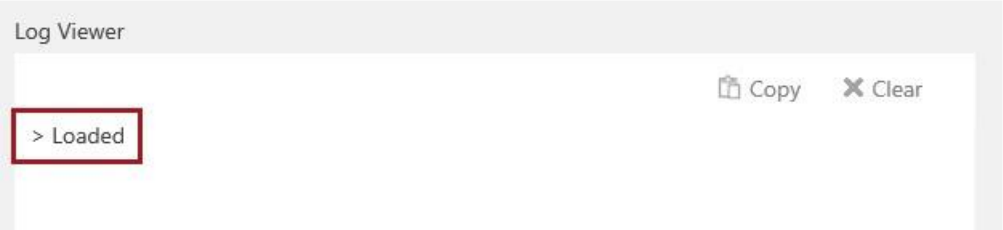
5. In the **Code** box, review the JavaScript code designed to use these values to embed the report into the lower pane.

6. To run the code to embed the report, in the **Code** box, at the top-right corner, click **Run**.



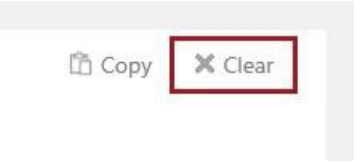
7. In the lower pane, ensure that your report loads.

This will be reported by feedback loaded in the log viewer.



```
> Loaded
```

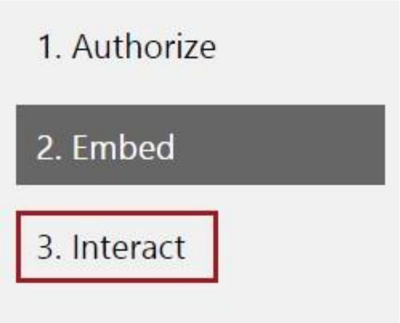
8. To clear the log comments, in the **Log Viewer** box, at the top-tight corner, click **Clear**.



Exploring the JavaScript API Capabilities

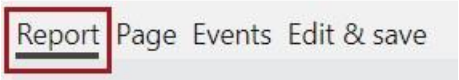
In this task, you will explore several capabilities supported by the JavaScript API.

1. Navigate to the **Interact** page.

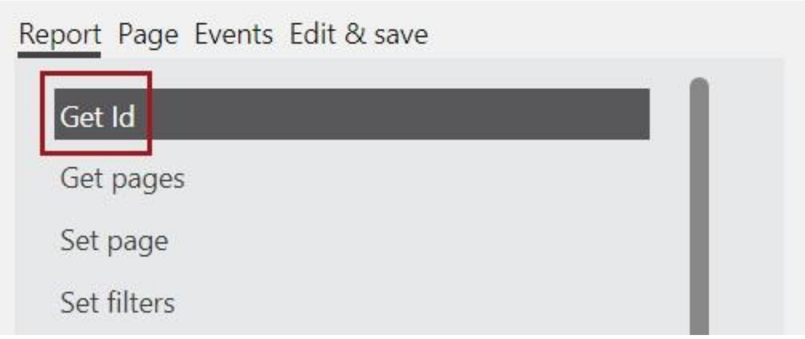


It is important to understand that all the programmatic operations you work with in this task are achieved by client-side script.

2. Above the list of operations, notice that the list is filtered by **Report** operations.

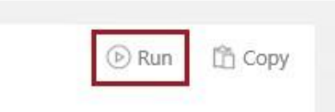


3. Notice also that the **Get Id** operation is selected.

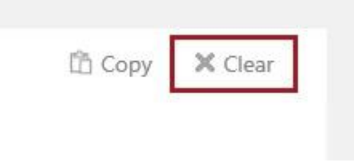


4. In the **Code** box, review the JavaScript code designed to perform the selected operation.

5. To run the code, in the **Code** box, at the top-right corner, click **Run**.



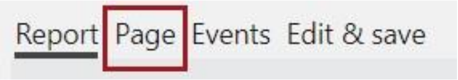
6. In the **Log Viewer** box, review the output, and then click **Clear**.



7. Work through the following list of operations, being sure to clear the log viewer between each.

Operation	Directions
Get pages	
Get filters	You can open the **Filters** pane, and filter by several regions *You will programmatically filter the report in the next exercise*
Remove filters	In the **Filters** pane, notice that the **Region** filter has been removed
Print	Notice that the browser print workflow is triggered
Update settings	Notice that the **Filters** pane is removed
Reload	The report is reloaded (useful if the report definition has changed, or the embed token has expired)
Refresh	The report data is refreshed
Full screen	In full screen mode, expand the **Filters** pane. Press **Escape** to exit this mode.

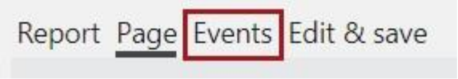
8. Select **Page** operations.



9. Work through the following list of operations, being sure to clear the log viewer between each.

Operation	Directions
Set active	Notice that the second page (**State Sales Analysis**) is now active
Get filters	(No page-level filters have been defined in the report)

10. Select **Events** operations.



11. Work through the following list of operations.

Operation	Directions
Page changed	Once running, use the report page tabs to make the first report page active
Data selected	Once running, change slicer selections, and highlight visual elements (columns and bars). Review the JSON document returned describing the data point.

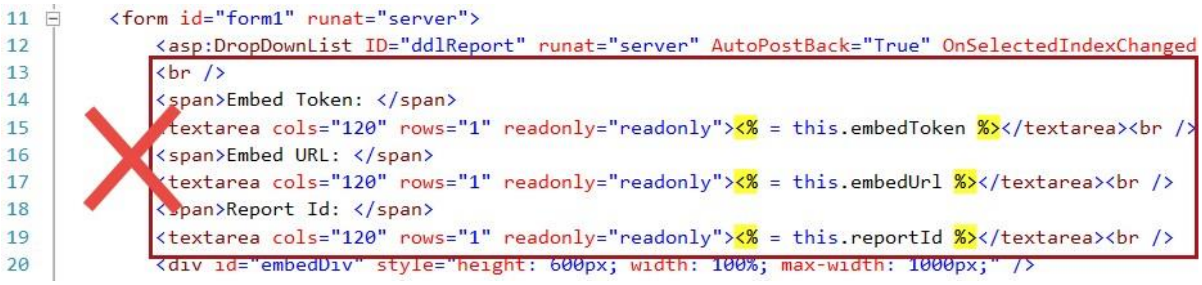
12. Close the sample web application window.

13. To stop debugging, close the **PowerBIEmbedding** application web browser.

Removing the Embed Values Output

In this task, you will remove the embed values output from the web form.

1. In the **EmbedReport.aspx** item, remove the HTML elements added in the first task of this exercise.



```
    <form id="form1" runat="server">
        <asp:DropDownList ID="ddlReport" runat="server" AutoPostBack="True" OnSelectedIndexChanged
        <br />
        <span>Embed Token: </span>
        <textarea cols="120" rows="1" readonly="readonly"><% = this.embedToken %></textarea><br />
        <span>Embed URL: </span>
        <textarea cols="120" rows="1" readonly="readonly"><% = this.embedUrl %></textarea><br />
        <span>Report Id: </span>
        <textarea cols="120" rows="1" readonly="readonly"><% = this.reportId %></textarea><br />
        <div id="embedDiv" style="height: 600px; width: 100%; max-width: 1000px;" />
```

2. Save the **EmbedReport.aspx** item.

Adding Client-Side Filtering

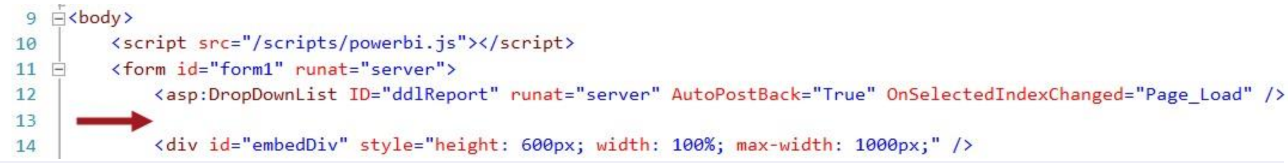
In this exercise, you will add functionality to the **EmbedReport.aspx** web form by using the Power BI JavaScript API to filter the embedded report by all regions, or a specific region.

Adding a Filter Dropdown List

In this task, you will modify the **EmbedReport.aspx** web form by adding a filter dropdown list.

1. To add a dropdown list, beneath the **asp:DropDownLists** element, add the following HTML elements:

   *For convenience, the elements can be copied from the **<CourseFolder>\PowerBIDevIAD\Lab02B\Assets\Snippets.txt** file.*



```
<body>
    <script src="/scripts/powerbi.js"></script>
    <form id="form1" runat="server">
        <asp:DropDownList ID="ddlReport" runat="server" AutoPostBack="True" OnSelectedIndexChanged="Page_Load" />

        <div id="embedDiv" style="height: 600px; width: 100%; max-width: 1000px;" />
```

HTML

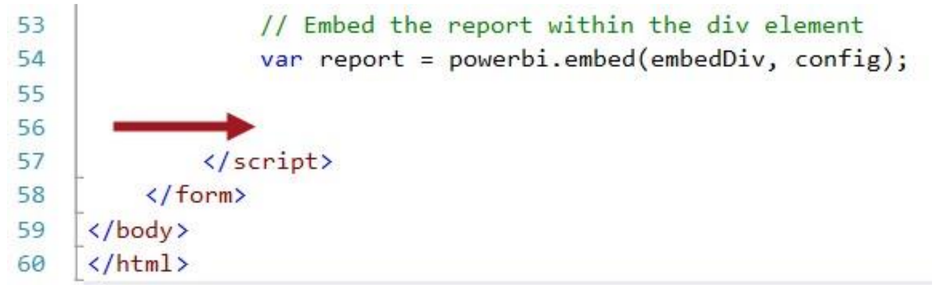
```
&nbsp;
<span>Region: </span>
<select id="ddlFilterRegion" onchange="filterRegion(); return false;">
 <option selected="selected" value="*">(All Regions)</option>
 <option value="Midwest">Midwest</option>
 <option value="New England">New England</option>
 <option value="Northeast">Northeast</option>
 <option value="Pacific Northwest">Pacific Northwest</option>
 <option value="Southern">Southern</option>
 <option value="Southwest">Southwest</option>
</select>
```

*The dropdown list presents each region, with the first item representing all regions. The **onchange** event hander will invoke the **filterRegion** function that you will add to the **script** element in the next task.*

Adding Client-Side Filter Logic

In this task, you will add the **filterRegion** function that will perform client-side filter logic.

1. To add the **filterRegion** function, inside the **script** element, add the following JavaScript function:



```
            // Embed the report within the div element
            var report = powerbi.embed(embedDiv, config);


        </script>
    </form>
</body>
</html>
```

JavaScript

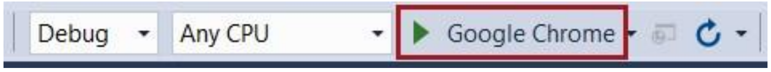
```
function filterRegion() {
  var report = powerbi.embeds[0];  var ddl =
document.getElementById("ddlFilterRegion");  var region
= ddl.options[ddl.selectedIndex].value;
  if (region == "*") {
        report.removeFilters()
                   .catch(error => { console.log(error); });
           return;
 }
  const basicFilter = {
           "$schema": "http://powerbi.com/product/schema#basic",
           "target": {
                   "table": "Region",
                   "column": "Region"
           },
           "operator": "In",
           "values": [
                   region
           ]
 }
  report.setFilters([basicFilter])
        catch(error => { console.log(error); }); }
```

*The function removes the region filter if **(All Regions)** is selected, otherwise it filters by the selected region.*

Reviewing the Filter Functionality

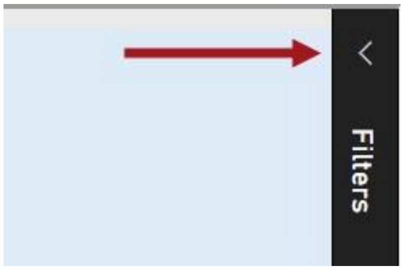
In this task, you will review the filter functionality.

1. On the toolbar, start the web application (start debugging).

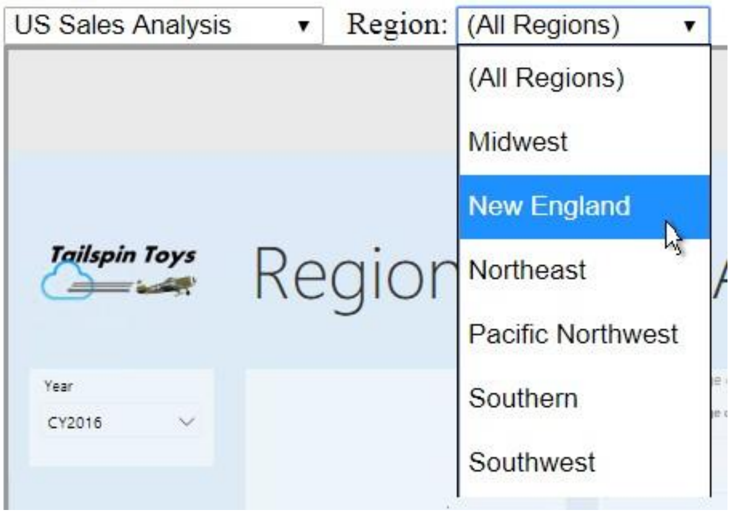


2. When the web browser opens, navigate to the **Embed Reports** page.

3. In the **US Sales Analysis** report, open the **Filters** pane.

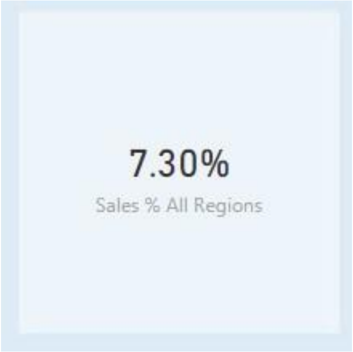


4. In the **Region** dropdown list, select any **Region**.

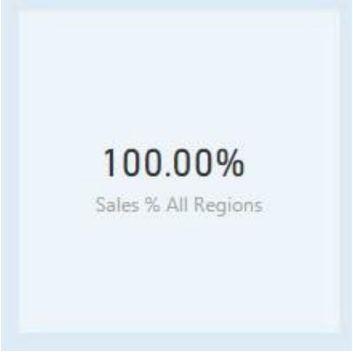


5. Notice that the selected region is applied as a filter.

6. Notice that the **Sales % All Regions** card displays a value less than 100%, representing the contribution the selected region made over all regions.



7. In the **Region** dropdown list, select any **(All Regions)**.
8. Notice that the report filter has been removed.
9. Notice that the **Sales % All Regions** card displays a value of 100%, as expected.

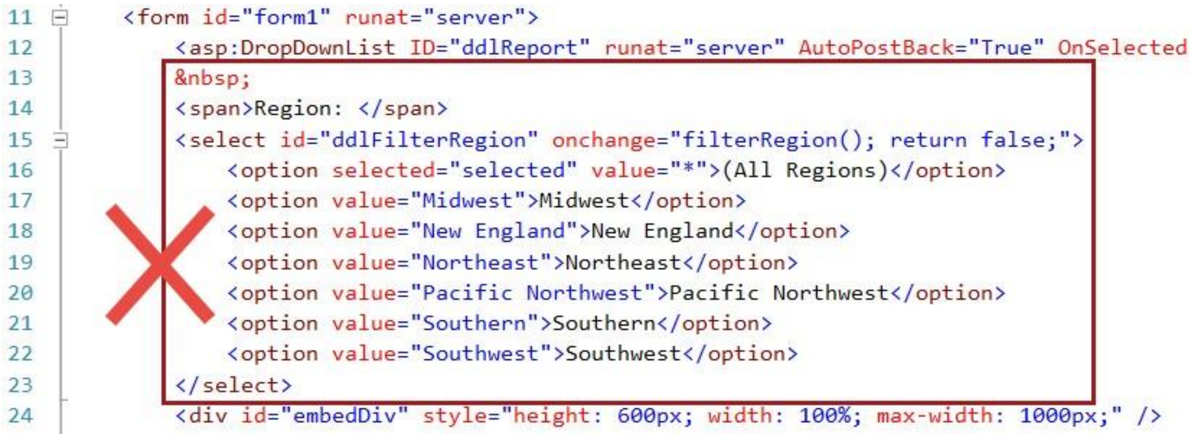


10. To stop debugging, close the **PowerBIEmbedding** application web browser.

Removing the Client-Side Filter Logic

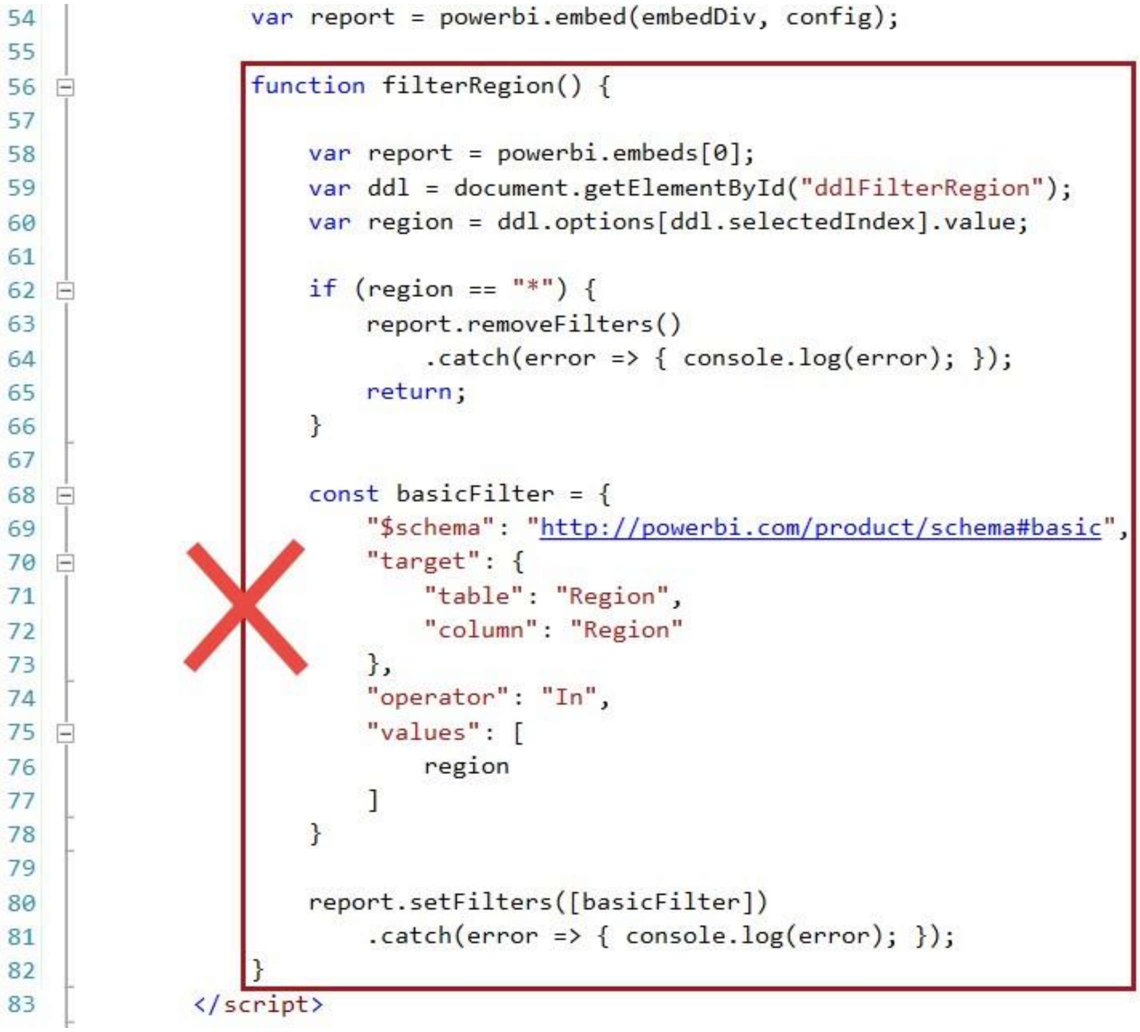
In this task, you will remove the client-side filter logic from the web form.

1. In the **EmbedReport.aspx** item, remove the HTML elements added in the first task of this exercise.



```
    <form id="form1" runat="server">
        <asp:DropDownList ID="ddlReport" runat="server" AutoPostBack="True" OnSelected
        &nbsp;
        <span>Region: </span>
        <select id="ddlFilterRegion" onchange="filterRegion(); return false;">
            <option selected="selected" value="*">(All Regions)</option>
            <option value="Midwest">Midwest</option>
            <option value="New England">New England</option>
            <option value="Northeast">Northeast</option>
            <option value="Pacific Northwest">Pacific Northwest</option>
            <option value="Southern">Southern</option>
            <option value="Southwest">Southwest</option>
        </select>
        <div id="embedDiv" style="height: 600px; width: 100%; max-width: 1000px;" />
```

2. In the **script** element, remove the **filterRegion** function.



```
            var report = powerbi.embed(embedDiv, config);

            function filterRegion() {

                var report = powerbi.embeds[0];
                var ddl = document.getElementById("ddlFilterRegion");
                var region = ddl.options[ddl.selectedIndex].value;

                if (region == "*") {
                    report.removeFilters()
                        .catch(error => { console.log(error); });
                    return;
                }

                const basicFilter = {
                    "$schema": "http://powerbi.com/product/schema#basic",
                    "target": {
                        "table": "Region",
                        "column": "Region"
                    },
                    "operator": "In",
                    "values": [
                        region
                    ]
                }

                report.setFilters([basicFilter])
                    .catch(error => { console.log(error); });
            }
        </script>
```

3. Save the **EmbedReport.aspx** item.

*You will modify the solution to apply row-level security (RLS) to enforce data permissions in **Lab 02C**.*

Summary

In this lab, you explored the sample web application and various Power BI JavaScript API operations. You then added client-side filtering to the **EmbedReport.aspx** web form to enable filtering by region.