

Deep Dive into the Power BI JavaScript API



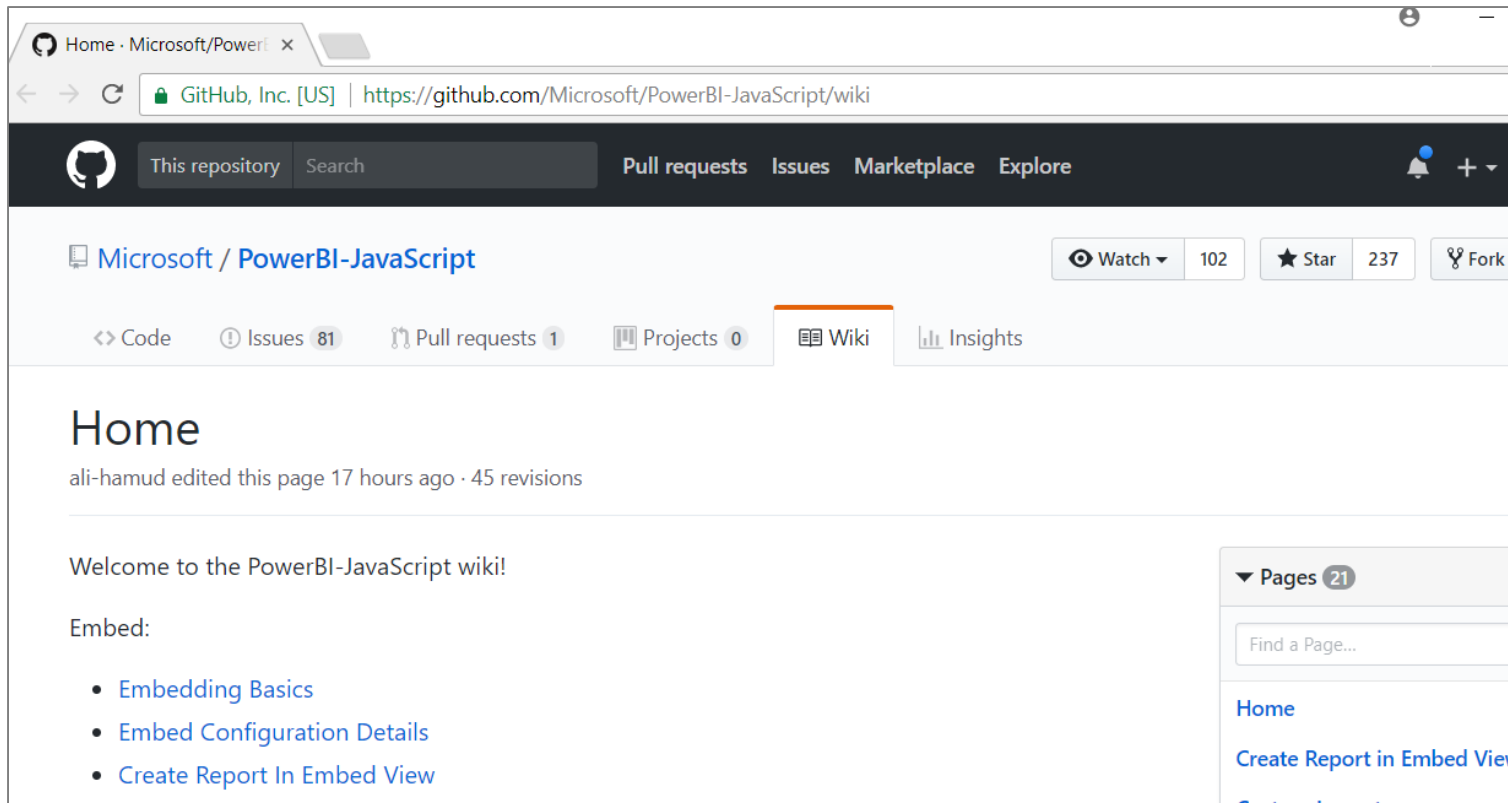
Agenda

- Embedding Dashboards, Dashboard Tiles and the Q&A Experience
- Embedding a New Report with Save-As Redirect Logic
- Setting Report Filters and Applying Bookmarks
- Embedding Reports using Phased Loading
- Embedding Report Visuals using a Custom Layout
- Extending an Embedded Report with Custom Menu Commands



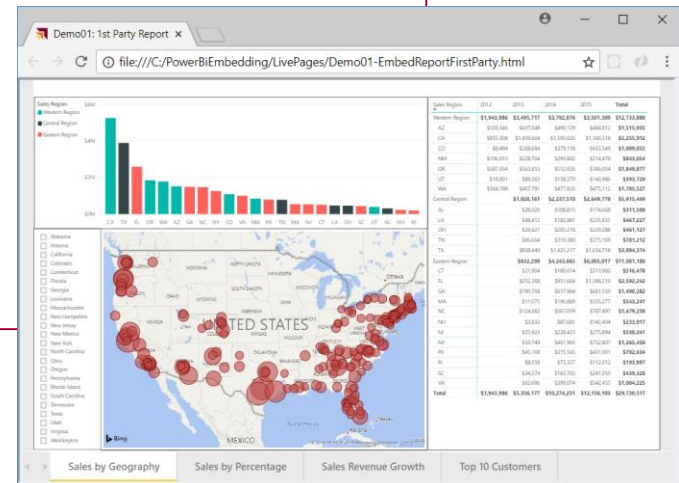
Power BI JavaScript API (powerbi.js)

- Power BI JavaScript API used to embed resources in browser
 - GitHub repo at <https://github.com/Microsoft/PowerBI-JavaScript/wiki>
 - GitHub repository contains code, docs, wiki and issues list



Hello World with Power BI Embedding

- **powerbi.js** library provides **powerbi** as top-level service object
 - You call **powerbi.embed** and pass **configuration** object with access token
 - **models** object available to supply configuration settings
 - **configuration** object sets **tokenType** to either **TokenType.Embed** or **TokenType.Aad**

[illegible]

Embedded Report Configuration Options

- **permissions** determines what permissions are given to user on resource
- **viewMode** determines when report opens in read-only view or edit view
- **pageView** determines how reports scales to fit embed container element

```
var models = window['powerbi-client'].models;  
  
var config: embed.IEmbedConfiguration = {  
  type: 'report',  
  id: reportId,  
  embedUrl: embedUrl,  
  accessToken: accessToken,  
  tokenType: models.TokenType.Embed,  
  permissions: models.Permissions.All,  
  viewMode: models.ViewMode.Edit,  
  pageView: "fitToWidth",  
  settings: {  
    filterPaneEnabled: false,  
    navContentPaneEnabled: false  
  }  
};
```

Read: Allows view report only.
ReadWrite: Allows view, Edit and Save report.
Copy: Allows Save a copy of a report using Save As.
Create: Allows creating a new report.
All: Allows everything.

View - Opens report in View mode.
Edit - Opens report in Edit mode.

fitToWidth: Fit to width of host HTML element.
oneColumn: Opens in single column.
actualSize: Actual size as designed in report



PowerBiEmbeddedScratchpad Sample

<https://github.com/CriticalPathTraining/PowerBiEmbeddedScratchpad>

The screenshot shows the GitHub repository page for **CriticalPathTraining / PowerBiEmbeddedScratchpad**. The page includes a navigation bar with links to Features, Business, Explore, Marketplace, and Pricing. The repository name is displayed at the top, along with buttons for Watch (3), Star (0), and Fork (0). Below the repository name, there are tabs for Code, Issues (0), Pull requests (0), Projects (0), and Insights. The description of the repository is "A sample application for developer's learning about Power BI Embedding". The repository statistics show 4 commits, 1 branch, 0 releases, and 1 contributor. The license is MIT. The file list shows the following files and their last commit dates:

File	Commit	Time
TedPattison Updates	Latest commit 3d2e699	7 days ago
DemoPagesWithEmbedding	Updates	8 days ago
PBIX	Updates	7 days ago
PowerBiEmbeddedScratchpad	Updates	8 days ago



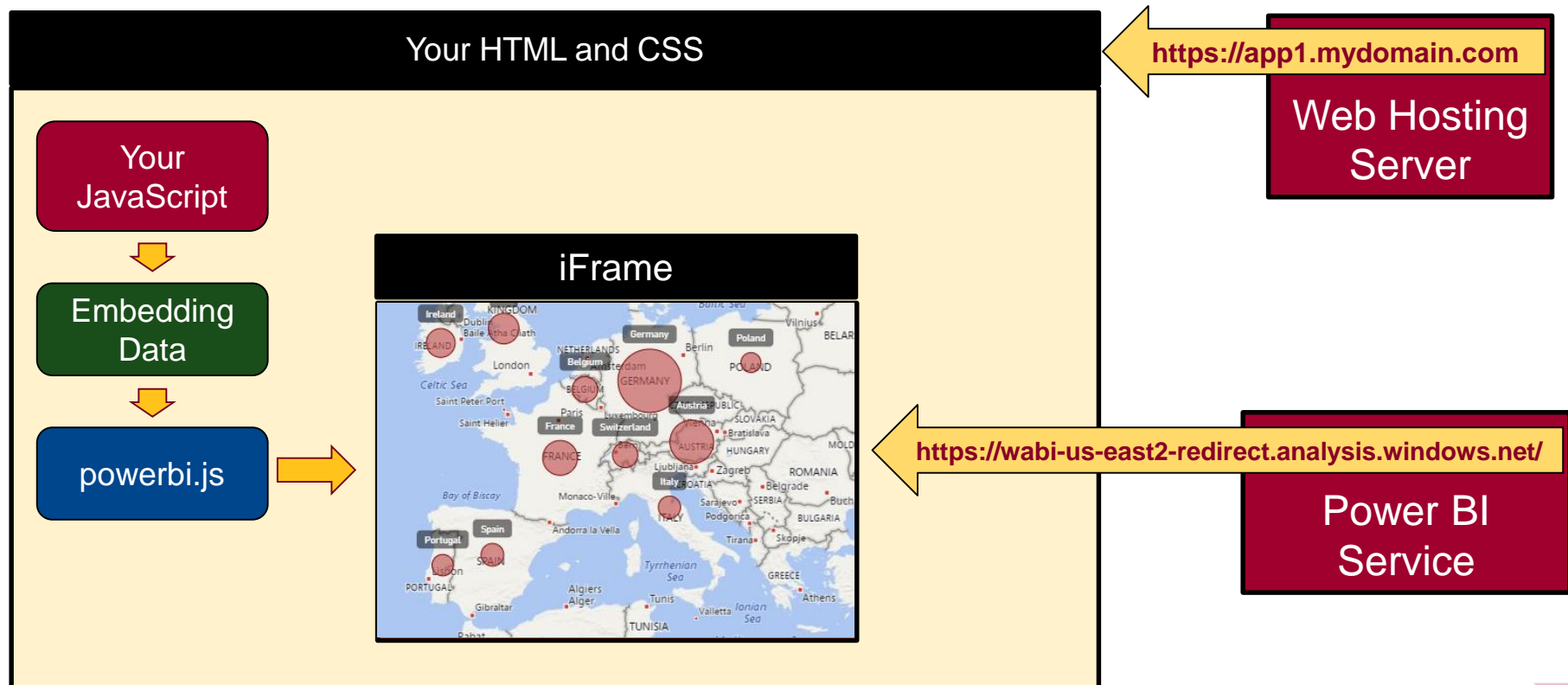


DEMO

The Power BI Embedded Scratchpad App

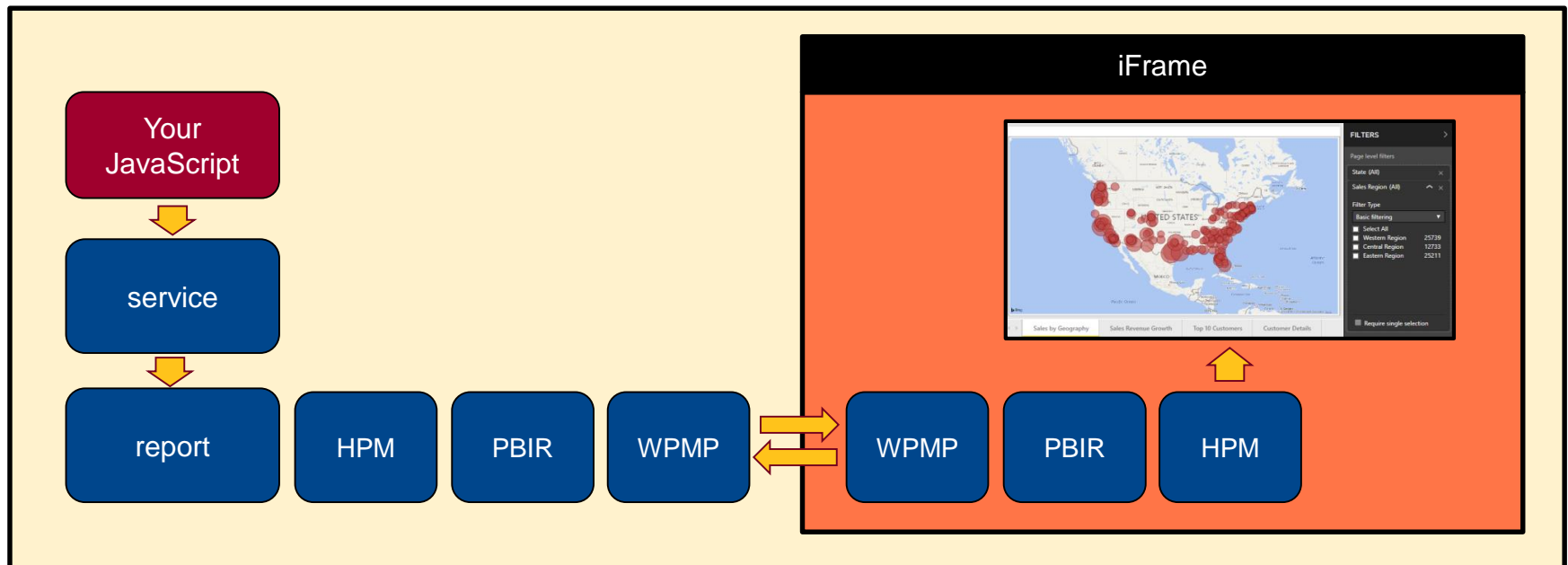
Report Embedding Architecture

- Embedding involves creating an iFrame on the page
 - PBIJS transparently creates iFrame and sets source to Power BI Service
 - ***The iFrame and hosting page originate from different DNS domains***



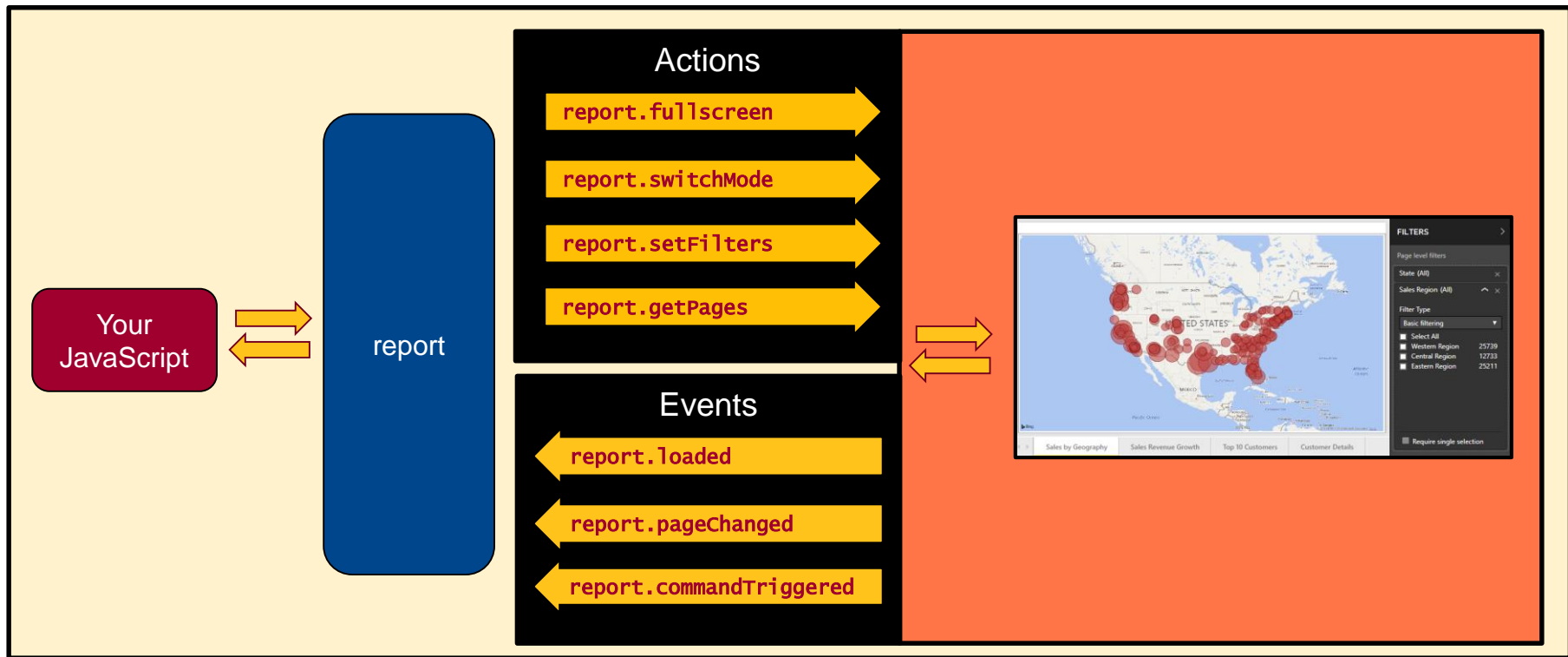
Post Message Communications Flow

- 4 extra libraries used communicate with report in iFrame
 - window-post-message-proxy (WPMP)
 - http-post-message (HPM)
 - powerbi-router (PBIR)
 - powerbi-models (PBIM)



A Promise-based Programming Model

- Design of PBIJS simulates HTTP protocol
 - Creates more intuitive programming model for developers
 - Programming based on asynchronous requests and promises
 - Embedded objects programmed using actions and events



Handling Report Events

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

var pages;

report.on('loaded', function () {
    // call getPages with callback
    report.getPages().then(
        function (reportPages) {
            pages = reportPages;
            // call method to load pages into nav menu
            loadReportPages(pages);
        });
});

var loadReportPages = function (pages) {
    for (var index = 0; index < pages.length; index++) {
        // determine which pages are visible and not hidden
        if (pages[index].visibility == 0) { // 0 means visible and 1 means hidden
            var reportPageDisplayName = pages[index].displayName;
            pageNavigation.append($("<li>")
                .append($("<a href='javascript:;'>")
                    .text(pages[index].displayName))
                .click(function (domEvent) {
                    var targetPageName = domEvent.target.textContent;
                    // get target page from pages collection
                    var targetPage = pages.find(function (page) { return page.displayName === targetPageName; }));
                    // navigate report to target page
                    targetPage.setActive();
                }));
        }
    }
}
```

```
▼ Report ⓘ
  ▼ allowedEvents: Array(12)
    0: "loaded"
    1: "saved"
    2: "rendered"
    3: "saveAsTriggered"
    4: "error"
    5: "dataSelected"
    6: "filtersApplied"
    7: "pageChanged"
    8: "commandTriggered"
    9: "swipeStart"
    10: "swipeEnd"
    11: "bookmarkApplied"
```



Embedding a New Report

```
// Get data required for embedding
var embedWorkspaceId= "7f4576c7-039a-472f-b998-546a572d5da2";
var embedDatasetId = "b4a48602-71da-42b2-8cf5-44d35b2ac70b";
var embedUrl = "https://app.powerbi.com/reportEmbed?groupId=7f4576c7-039a-472f-b998-546a5
var accessToken = "H4sIAAAAAAAEAB2Wxw60CA6E3-W_shIZmpXmQE5NztzIOwdG--7bM3dbsj67qvz3HzN5-i

// Get models object to access enums for embed configuration
var models = window['powerbi-client'].models;

var config = {
  datasetId: embedDatasetId,
  embedUrl: embedUrl,
  accessToken: accessToken,
  tokenType: models.TokenType.Embed,
};

// Get a reference to the embedded report HTML element
var embedContainer = document.getElementById('embedContainer');

// Embed the report and display it within the div container.
var report = powerbi.createReport(embedContainer, config);
```



New Report with SaveAs Redirect

```
// Embed the report and display it within the div container.
var newReport = powerbi.createReport(embedContainer, config);

// this event fires whenever user runs save or SaveAs command on a new report
newReport.on("saved", function (event) {

    // get ID and name of new report
    var newReportId = event.detail.reportObjectId;
    var newReportName = event.detail.reportName;

    // set new title for browser window
    document.title = newReportName;

    // reset report container element
    powerbi.reset(embedContainer);

    config = {
        type: 'report',
        id: newReportId,
        embedUrl: "https://app.powerbi.com/reportEmbed?reportId=" + newReportId + "&groupId=" + embedWorkspaceId,
        accessToken: accessToken,
        tokenType: models.TokenType.Aad,
        permissions: models.Permissions.All,
        viewMode: models.ViewMode.Edit,
    };

    // Embed the report and display it within the div container.
    var savedReport = powerbi.embed(embedContainer, config);
```



Embedding the Q&A Experience

```
// Get data required for embedding
var datasetId = "b4a48602-71da-42b2-8cf5-44d35b2ac70b";
var embedUrl = "https://app.powerbi.com/qnaEmbed?groupId=7f4576c7-039a-472f-b998-546a57";
var accessToken = "H4sIAAAAAAAEAC2Wx6rFDI6E3-XfeuA4h4Fe00ecvXPO0buZd58L3XuBpK8K1f79j5W-";

// Get models object to access enums for embed configuration
var models = window['powerbi-client'].models;

var config = {
  type: 'qna',
  tokenType: models.TokenType.Embed,
  accessToken: accessToken,
  embedUrl: embedUrl,
  datasetIds: [ datasetId ],
  viewMode: models.QnaMode.Interactive,
  question: "What is sales revenue by quarter and sales region as stacked area chart"
};

var embedContainer = document.getElementById('embedContainer');

var embeddedObject = powerbi.embed(embedContainer, config);
```



Configuration Options

```
// Get models object to access enums for embed configuration
var models = window['powerbi-client'].models;

var config = {
  type: 'report',
  id: embedReportId,
  embedUrl: embedUrl,
  accessToken: accessToken,
  tokenType: models.TokenType.Embed,
  permissions: models.Permissions.All,
  viewMode: models.ViewMode.View,
  settings: {
    filterPaneEnabled: false,
    navContentPaneEnabled: true,
    localeSettings: { language: "en", formatLocale: "es" },
    background: models.BackgroundType.Transparent
  }
};

// Get a reference to the embedded report HTML element
var reportContainer = document.getElementById('embedContainer');

// Embed the report and display it within the div container.
var report = powerbi.embed(reportContainer, config);
```



Summary

- ✓ Embedding Dashboards, Dashboard Tiles and the Q&A Experience
- ✓ Embedding a New Report with Save-As Redirect Logic
- ✓ Setting Report Filters and Applying Bookmarks
- ✓ Embedding Reports using Phased Loading
- ✓ Embedding Report Visuals using a Custom Layout
- ✓ Extending an Embedded Report with Custom Menu Commands

