

Embedding Power BI Reports with ASP.NET MVC



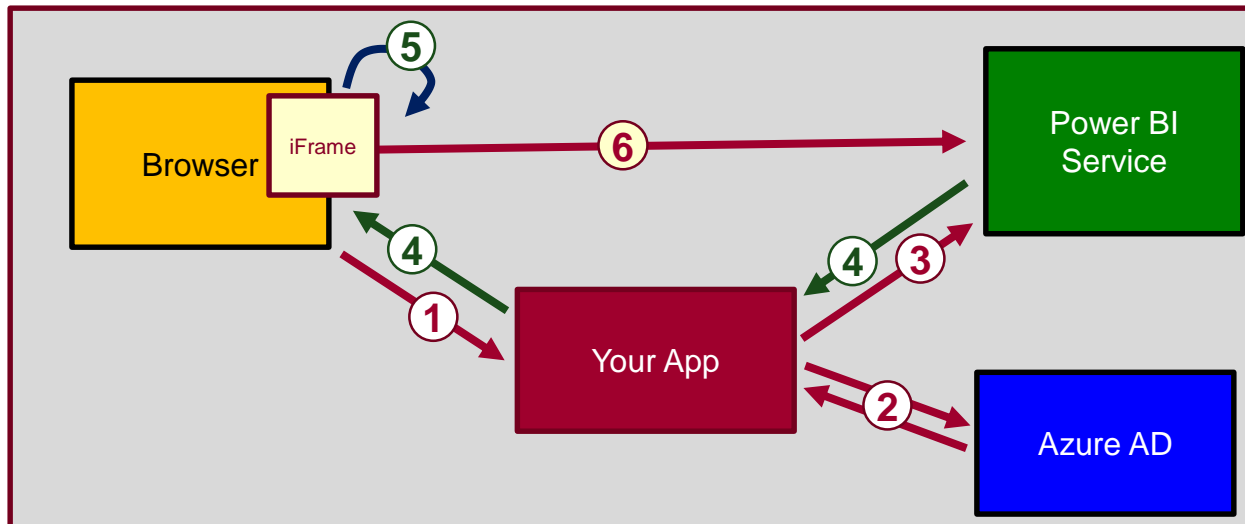
Agenda

- Power BI Embedding Onboarding Experience
- Creating ASP.NET MVC Project for Power BI Embedding
- Retrieving Power BI Embedding Data
- Generating and Managing Embed Tokens
- Writing Client-side Code to Embed Resources



Power BI Embedding – The Big Picture

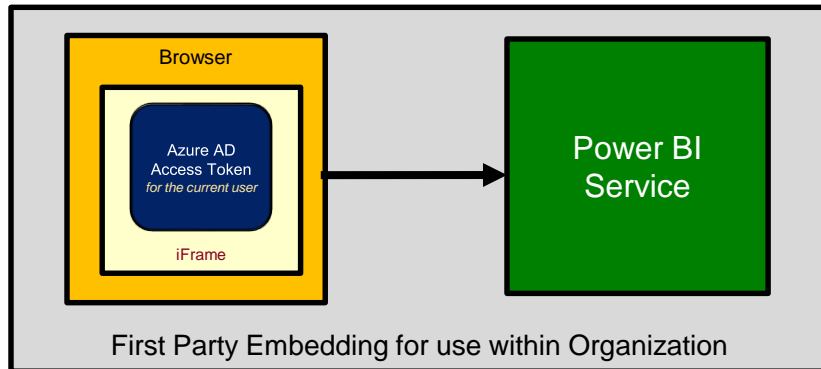
- User launches your app using a browser
- App authenticates with Azure Active Directory and obtains access token
- App uses access token to call to Power BI Service API
- App retrieves data for embedded resource and passes it to browser.
- Client-side code uses Power BI JavaScript API to create embedded resource
- Embedded resource session created between browser and Power BI service



First Party Embedding vs Third Party Embedding

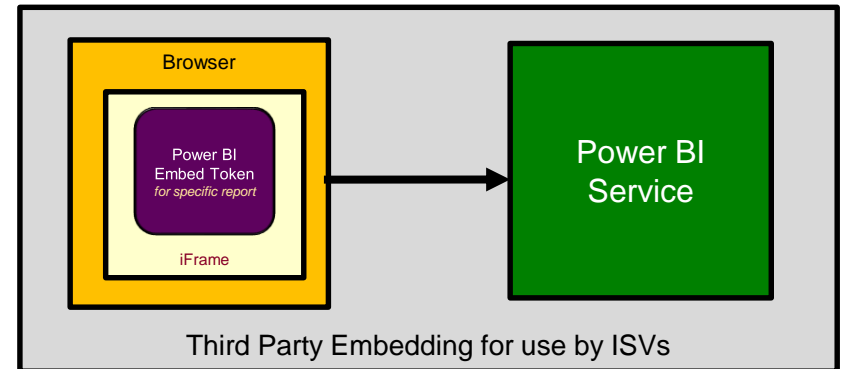
First Party Embedding

- Known as **User-Owns-Data** Model
- All users require a Power BI license
- Useful in corporate environments
- App authenticates as current user
- Your code runs with user's permissions
- User's access token passed to browser



Third Party Embedding

- Known as **App-Owns-Data** Model
- No users require Power BI license
- Useful for commercial applications
- App authenticates with master user account
- Your code runs with admin permissions
- Embed token passed to browser



Power BI Embedding Onboarding Experience

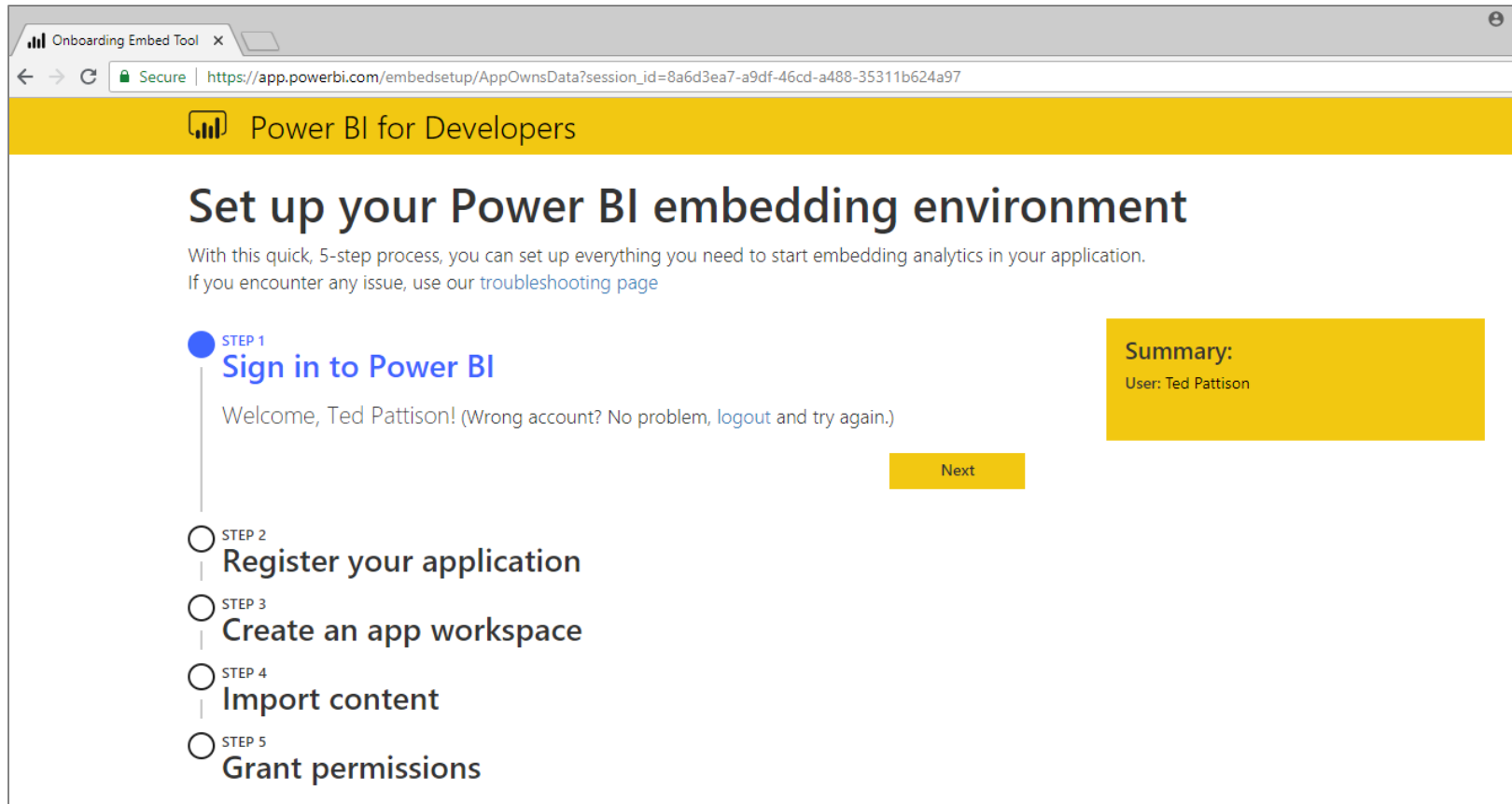
- <https://app.powerbi.com/embedsetup>

The screenshot shows the 'Onboarding Embed Tool' in a web browser. The page title is 'Power BI for Developers' and the main heading is 'Set up your Power BI embedding environment'. Below this, it says 'Choose whether you'd like to embed Power BI in an application for your customers or in an application, website, or portal for your organization's internal users.' and 'Choose an embedding solution'. There are two main options:

- Embed Power BI in your product for your customers—for ISVs and developers**
Users will not need a Power BI account to view and interact with embedded data.
This option is associated with the button 'Embed for your customers' and is labeled **Third Party Embedding** with a red arrow pointing to it.
- Embed Power BI for your organization's internal users—for enterprises**
Users will need a Power BI account and permission to access the underlying content to view and interact with embedded data.
This option is associated with the button 'Embed for your organization' and is labeled **First Party Embedding** with a red arrow pointing to it.



App-Owns-Data Experience – Step 1



The screenshot shows a web browser window with the title "Onboarding Embed Tool". The address bar shows a secure connection to https://app.powerbi.com/embedsetup/AppOwnsData?session_id=8a6d3ea7-a9df-46cd-a488-35311b624a97. The page header is "Power BI for Developers". The main heading is "Set up your Power BI embedding environment". Below this, a subheading states: "With this quick, 5-step process, you can set up everything you need to start embedding analytics in your application. If you encounter any issue, use our [troubleshooting page](#)".

The onboarding process is shown as a vertical list of five steps:

- STEP 1** (Active): **Sign in to Power BI**
Welcome, Ted Pattison! (Wrong account? No problem, [logout](#) and try again.)
[Next](#)
- STEP 2**: Register your application
- STEP 3**: Create an app workspace
- STEP 4**: Import content
- STEP 5**: Grant permissions

A yellow summary box on the right contains the text: "Summary: User: Ted Pattison".



App-Owns-Data Experience – Step 2

STEP 2
Register your application

Register your application with Azure AD to allow your application access to the Power BI REST APIs and to set resource permissions for your app. You can change this later in the Microsoft Azure portal. [Learn more](#)

Application Name

Enter a display name to identify your application in Azure

API access

Select the APIs and the level of access your app needs. You can change these settings later in the Azure portal. [Learn more](#)

☐ Select All

Read APIs ⓘ	Write APIs ⓘ	Create APIs ⓘ
<input checked="" type="checkbox"/> Read All Datasets	<input type="checkbox"/> Read and Write All Datasets	<input type="checkbox"/> Create Content
<input checked="" type="checkbox"/> Read All Dashboards	<input type="checkbox"/> Read and Write All Dashboards	
<input checked="" type="checkbox"/> Read All Reports	<input type="checkbox"/> Read and Write All Reports	
<input checked="" type="checkbox"/> Read All Groups	<input type="checkbox"/> Read and Write All Workspaces	
<input checked="" type="checkbox"/> Read All Workspaces	<input type="checkbox"/> Read and Write All Capacities	
<input checked="" type="checkbox"/> Read All Capacities		

By clicking Register, you agree to the [terms of use](#)

Register



What Just Happened?

- The Onboarding Tool created a new Azure AD App
 - Configured as a Native app as opposed to a Web app / API

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation links for 'Create a resource', 'All services', and 'FAVORITES'. The main content area is titled 'critical path training labs - App registrations' and shows a list of application registrations. The 'My First Power BI Embedding App' is highlighted with a dashed blue box and an arrow pointing to it. The right-hand pane shows the 'DELEGATED PERMISSIONS' table, which lists various permissions and their status.

DELEGATED PERMISSIONS		REQUIRES ADMIN
<input type="checkbox"/>	Read and write all dataflows	No
<input type="checkbox"/>	View all dataflows	No
<input type="checkbox"/>	Read and write all content in tenant	Yes
<input type="checkbox"/>	Read and Write all Reports	No
<input checked="" type="checkbox"/>	View users Groups	No
<input type="checkbox"/>	View all Groups	No
<input checked="" type="checkbox"/>	View all Reports (preview)	No
<input type="checkbox"/>	Create content (preview)	No
<input type="checkbox"/>	View content properties (preview)	No
<input type="checkbox"/>	Read and Write all Datasets	No
<input checked="" type="checkbox"/>	View all Datasets	No
<input checked="" type="checkbox"/>	View all Dashboards (preview)	No
<input type="checkbox"/>	Add data to a user's dataset (preview)	No
<input type="checkbox"/>	Read and Write all Dashboards	No
<input type="checkbox"/>	View all content in tenant	Yes
<input type="checkbox"/>	Read and write all workspaces	No
<input checked="" type="checkbox"/>	View all workspaces	No
<input type="checkbox"/>	Read and write all capacities	No
<input checked="" type="checkbox"/>	View all capacities	No

App-Owns-Data Experience – Step 3

STEP 1
✓ Sign in to Power BI

STEP 2
✓ Register your application

STEP 3
● Create an app workspace

STEP 4
○ Import content

STEP 5
○ Grant permissions

In order to start adding content to Power BI, you need to create an app workspace. You can manage your app workspace in the Power BI web service. [Learn more](#)

Name your app workspace

Create app workspaceSkip

Summary:

User: Ted Pattison

Application: My First Power BI Embedding App

Application ID:
122e858a-1b70-40f8-aa5e-de0ff4f6fca6



App-Owns-Data Experience – Step 4

STEP 1
| Sign in to Power BI

STEP 2
| Register your application

STEP 3
| Create an app workspace

STEP 4
| **Import content**

STEP 5
| Grant permissions

To embed Power BI content in your app, you can use a sample report to get started, or upload your own .pbix file.

☒ Sample Power BI report

☐ Upload .pbix file

Summary:

User: Ted Pattison

Application: My First Power BI Embedding App

Application ID:

122e858a-1b70-40f8-aa5e-de0ff4f6fca6

Workspace: My New App Workspace

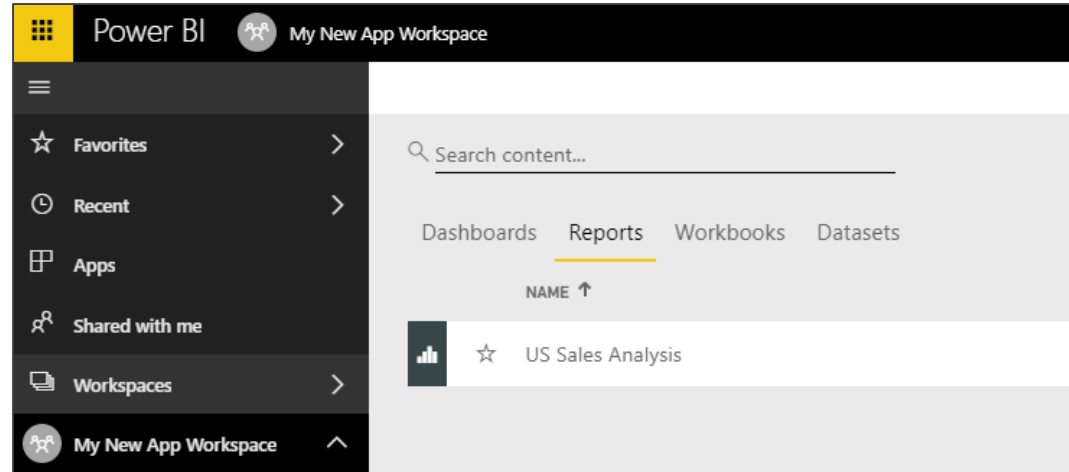
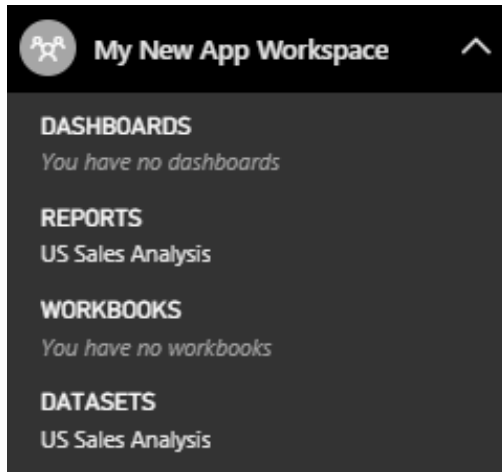
Workspace ID:

ed3bd652-89c9-4262-8ac0-4757b38ebf69



What Just Happened?

- The Onboarding Tool created a new Power BI App Workspace
 - It also uploaded a PBIX file so there is already one dataset and one report
 - New app workspace is not automatically associated with a dedicated capacity



App-Owns-Data Experience – Step 5

- ✓ STEP 1
| Sign in to Power BI
- ✓ STEP 2
| Register your application
- ✓ STEP 3
| Create an app workspace
- ✓ STEP 4
| Import content
- STEP 5
Grant permissions

Grant permissions for your application to access the selected APIs with the signed in account.
[Learn how to grant permissions directly in Azure Portal.](#)

Grant permissions

Summary:

User: Ted Pattison

Application: My First Power BI Embedding App

Application ID:

122e858a-1b70-40f8-aa5e-de0ff4f6fca6

Workspace: My New App Workspace

Workspace ID:

ed3bd652-89c9-4262-8ac0-4757b38ebf69

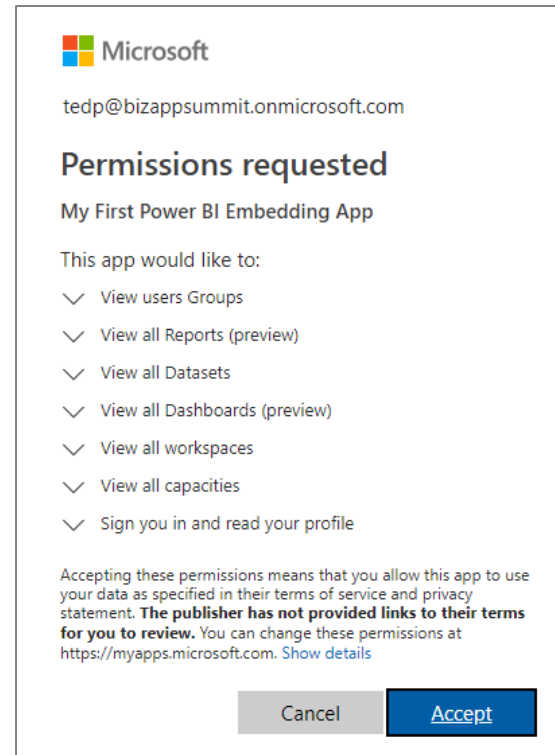
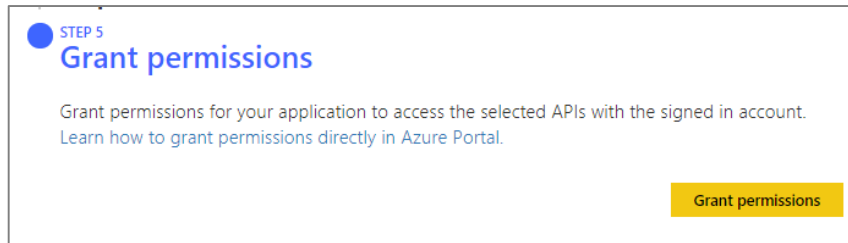
Report ID:

a5c37d6c-f26d-4565-a5e7-19fe1a3fcb79



Granting Permissions to the App

- Users must consent to permissions requested by an app



All Great Experiences Must Come to an End



Power BI for Developers

Set up your Power BI embedding environment

Success! Your Power BI embedding environment is ready to use.

Download the sample app to explore your embedded content with all the relevant information from the environment you've set up, pre-configured and ready to test.

If you encounter any issue, use our [troubleshooting page](#)

Download Sample App



Power BI Developer Center

Find all the resources you need in one place, including documentation, APIs, tutorials, and more!



Summary:

User: Ted Pattison

Application: My First Power BI Embedding App

Application ID:

122e858a-1b70-40f8-aa5e-de0ff4f6fca6

Workspace: My New App Workspace

Workspace ID:

ed3bd652-89c9-4262-8ac0-4757b38ebf69

Report ID:

a5c37d6c-f26d-4565-a5e7-19fe1a3fcb79

Download Sample App



User-Owns-Data Onboarding Experience

- Users-Owns-Data Experience is similar to App-Owns-Data experience
 - Users-Owns-Data Experience allows you to create app a **Native** or **Web app / API**
 - App create as Web app / API must be created with valid Reply URL
 - Process configures Azure app with a secret key known as Application secret
 - Application secret allows app to authenticate using authorization code grant flow

Summary:

User: Ted Pattison

Application: My User-Owns-Data App

Application ID:

70c57777-bb2f-4079-bc4d-aca03c51dc78

Application secret:

ASXQUJIO2AaLuwoIEbEbSXCOx/+4Km45MJ...

Workspace: My First Party App

Workspace ID:

cbe63eb7-bb17-4dae-8815-9159c9ae9859

Report ID:

19d3e036-1fd2-49f4-b218-a5330bf2f4c4

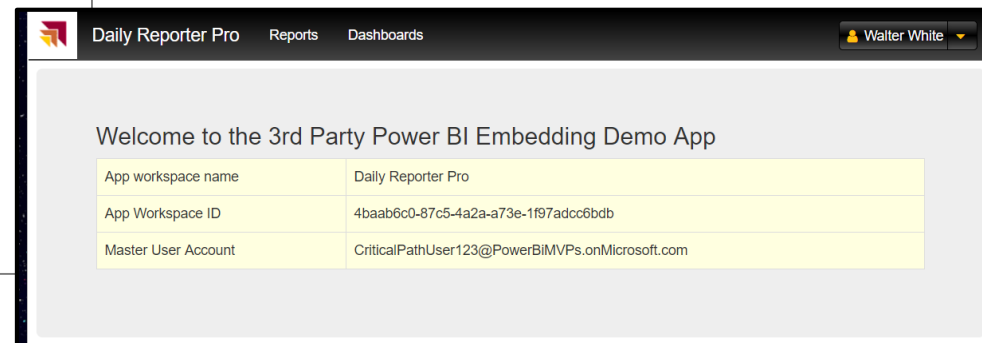
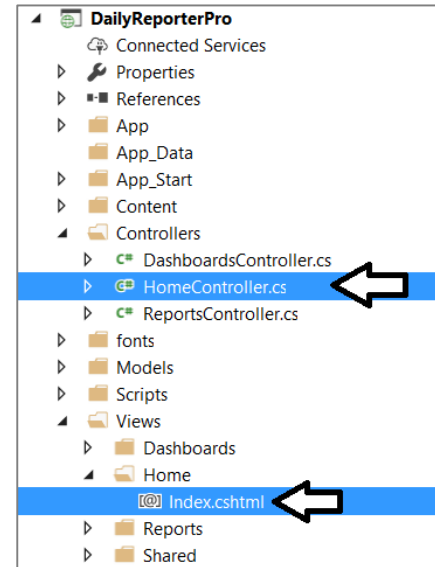
[Download Sample App](#)



MVC Controllers and Views

```
public class HomeController : Controller {  
  
    public async Task<ActionResult> Index() {  
        var viewModel = await PbiEmbeddingManager.GetHomeViewModel();  
        return View(viewModel);  
    }  
}
```

```
Index.cshtml  + x  
@model DailyReporterPro.Models.HomeViewModel  
  
<div id="home-view-container">  
    <div class="jumbotron">  
        <h3>Welcome to the 3rd Party Power BI Embedding Demo App</h3>  
  
        <table id="session-info-table" class="table table-bordered">  
            <tr>  
                <td>App workspace name</td>  
                <td>@Model.WorkspaceName</td>  
            </tr>  
            <tr>  
                <td>App Workspace ID</td>  
                <td>@Model.WorkspaceId</td>  
            </tr>  
            <tr>  
                <td>Master User Account</td>  
                <td>@Model.MasterUserAccount</td>  
            </tr>  
        </table>  
    </div>
```



Creating Model Classes for MVC Views

```
public class HomeViewModel {  
    public string WorkspaceName;  
    public string WorkspaceId;  
    public string MasterUserAccount;  
}
```

```
public static async Task<HomeViewModel> GetHomeViewModel() {  
    var client = GetPowerBiClient();  
    var workspaces = (await client.Groups.GetGroupsAsync()).Value;  
    var workspace = workspaces.Where(ws => ws.Id == appWorkspaceId).FirstOrDefault();  
    var viewModel = new HomeViewModel {  
        WorkspaceName = workspace.Name,  
        WorkspaceId = workspace.Id,  
        MasterUserAccount = pbiUserName  
    };  
    return viewModel;  
}
```



MVC View Models

```
namespace DailyReporterPro.Models {  
    public class HomeViewModel ...  
    public class DatasetViewModel ...  
    public class DatasetsViewModel ...  
    public class ReportViewModel ...  
    public enum ReportMode ...  
    public class ReportsViewModel ...  
    public class DashboardViewModel ...  
    public class DashboardsViewModel ...  
}
```

```
public static async Task<HomeViewModel> GetHomeViewModel() ...  
public static async Task<DatasetsViewModel> GetDatasetsViewModel() ...  
public static async Task<ReportsViewModel> GetReportsViewModel(string reportId, string datasetId) ...  
public static async Task<DashboardsViewModel> GetDashboardsViewModel(string dashboardId) ...
```



Report and Dataset Info

```
// data required for embedding a report
class ReportEmbeddingData {
    public string reportId;
    public string reportName;
    public string embedUrl;
    public string accessToken;
}

// data required for embedding a dashboard
class DashboardEmbeddingData {
    public string dashboardId;
    public string dashboardName;
    public string embedUrl;
    public string accessToken;
}

// data required for embedding a dashboard
class DashboardTileEmbeddingData {
    public string dashboardId;
    public string TileId;
    public string TileTitle;
    public string embedUrl;
    public string accessToken;
}
```

```
// data required for embedding a new report
class NewReportEmbeddingData {
    public string workspaceId;
    public string datasetId;
    public string embedUrl;
    public string accessToken;
}

// data required for embedding QnA experience
class QnaEmbeddingData {
    public string datasetId;
    public string embedUrl;
    public string accessToken;
}
```



Data for First Party Report Embedding

```
public static ReportEmbeddingData GetReportEmbeddingDataFirstParty() {  
    PowerBIClient pbiclient = GetPowerBIClient();  
  
    var report = pbiclient.Reports.GetReportInGroup(workspaceId, reportId);  
    var embedUrl = report.EmbedUrl;  
    var reportName = report.Name;  
    var accessToken = GetAccessToken();  
  
    return new ReportEmbeddingData {  
        reportId = reportId,  
        reportName = reportName,  
        embedUrl = embedUrl,  
        accessToken = accessToken  
    };  
}
```



Embed Tokens

- You can embed reports using master user AAD token, but...
 - You might want embed resource using more restricted tokens
 - You might want stay within the bounds of Power BI licensing terms
- You generate embed tokens with the Power BI Service API
 - Each embed token created for one specific resource
 - Embed token provides restrictions on whether user can view or edit
 - Embed token can only be generated in dedicated capacity (semi-enforced)
 - Embed token can be generated to support row-level security (RLS)

```
Report report = reports.Where(r => r.Id == reportId).First();  
var generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "edit");  
var token = client.Reports.GenerateTokenInGroupAsync(appWorkspaceId,  
                                                    report.Id,  
                                                    generateTokenRequestParameters).Result;
```



Data for Third Party Report Embedding

```
public static ReportEmbeddingData GetReportEmbeddingData() {  
    PowerBIClient pbiclient = GetPowerBIClient();  
  
    var report = pbiclient.Reports.GetReportInGroup(workspaceId, reportId);  
    var embedUrl = report.EmbedUrl;  
    var reportName = report.Name;  
  
    // create token request object  
    GenerateTokenRequest generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");  
  
    // call to Power BI Service API and pass GenerateTokenRequest object to generate embed token  
    string embedToken = pbiclient.Reports.GenerateTokenInGroup(workspaceId,  
                                                                report.Id,  
                                                                generateTokenRequestParameters).Token;  
  
    return new ReportEmbeddingData {  
        reportId = reportId,  
        reportName = reportName,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



Getting the Data for Dashboard Embedding

```
public static DashboardEmbeddingData GetDashboardEmbeddingData() {  
    PowerBIClient pbiclient = GetPowerBIClient();  
  
    var dashboard = pbiclient.Dashboards.GetDashboardInGroup(workspaceId, dashboardId);  
    var embedUrl = dashboard.EmbedUrl;  
    var dashboardDisplayName = dashboard.DisplayName;  
  
    GenerateTokenRequest generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");  
    string embedToken = pbiclient.Dashboards.GenerateTokenInGroup(workspaceId,  
                                                                    dashboardId,  
                                                                    generateTokenRequestParameters).Token;  
  
    return new DashboardEmbeddingData {  
        dashboardId = dashboardId,  
        dashboardName = dashboardDisplayName,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



Data for Dashboard Tile Embedding

```
public static DashboardTileEmbeddingData GetDashboardTileEmbeddingData() {  
    PowerBIClient pbiClient = GetPowerBiClient();  
  
    var tiles = pbiClient.Dashboards.GetTilesInGroup(workspaceId, dashboardId).Value;  
  
    // retrieve first tile in tiles connection  
    var tile = tiles[0];  
    var tileId = tile.Id;  
    var tileTitle = tile.Title;  
    var embedUrl = tile.EmbedUrl;  
  
    GenerateTokenRequest generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");  
    string embedToken = pbiClient.Tiles.GenerateTokenInGroup(workspaceId,  
                                                            dashboardId,  
                                                            tileId,  
                                                            generateTokenRequestParameters).Token;  
  
    return new DashboardTileEmbeddingData {  
        dashboardId = dashboardId,  
        TileId = tileId,  
        TileTitle = tileTitle,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



Data for New Report Embedding

```
public static NewReportEmbeddingData GetNewReportEmbeddingData() {  
    string embedUrl = "https://app.powerbi.com/reportEmbed?groupId=" + workspaceId;  
    PowerBIClient pbiClient = GetPowerBiClient();  
    GenerateTokenRequest generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "create",  
                                                                                        datasetId: datasetId);  
    string embedToken = pbiClient.Reports.GenerateTokenForCreateInGroup(workspaceId,  
                                                                           generateTokenRequestParameters).Token;  
    return new NewReportEmbeddingData {  
        workspaceId = workspaceId,  
        datasetId = datasetId,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



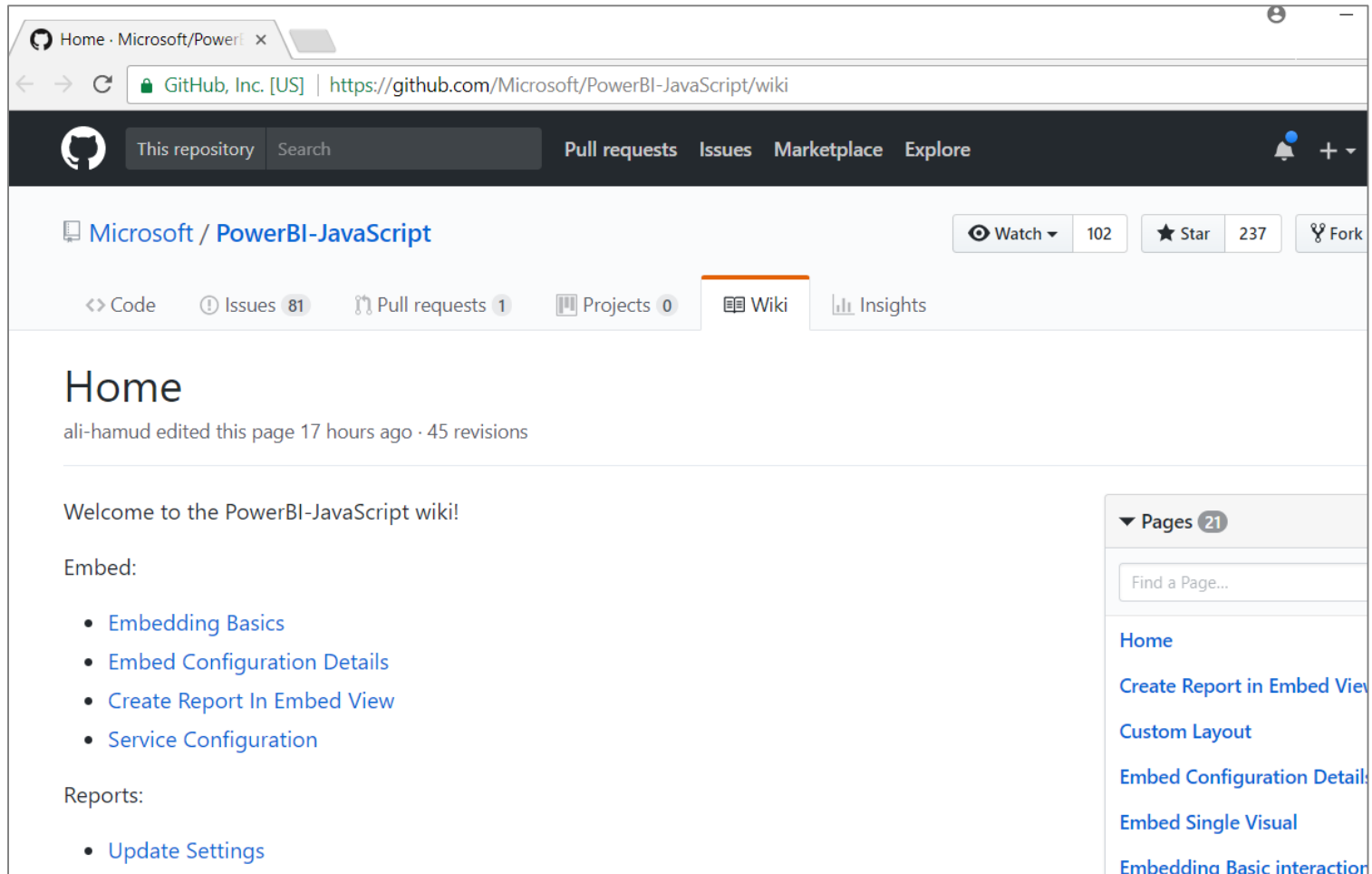
Data for Q&A Experience Embedding

```
public static QnaEmbeddingData GetQnaEmbeddingData() {  
    PowerBIClient pbiclient = GetPowerBIClient();  
  
    var dataset = pbiclient.Datasets.GetDatasetByIdInGroup(workspaceId, datasetId);  
  
    string embedUrl = "https://app.powerbi.com/qnaEmbed?groupId=" + workspaceId;  
    string datasetID = dataset.Id;  
  
    GenerateTokenRequest generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");  
  
    string embedToken = pbiclient.Datasets.GenerateTokenInGroup(workspaceId,  
                                                                dataset.Id,  
                                                                generateTokenRequestParameters).Token;  
  
    return new QnaEmbeddingData {  
        datasetId = datasetId,  
        embedUrl = embedUrl,  
        accessToken = embedToken  
    };  
}
```



Power BI JavaScript API (PBIJS)

- <https://github.com/Microsoft/PowerBI-JavaScript/wiki>



The screenshot shows a web browser window displaying the GitHub repository page for Microsoft/PowerBI-JavaScript. The browser's address bar shows the URL <https://github.com/Microsoft/PowerBI-JavaScript/wiki>. The repository page has a dark header with the GitHub logo, a search bar, and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the header, the repository name "Microsoft / PowerBI-JavaScript" is displayed, along with statistics: 102 Watchers, 237 Stars, and a Fork button. The "Wiki" tab is selected, showing a "Home" page. The page content includes a welcome message, a list of links for embedding Power BI reports, and a sidebar with a list of 21 wiki pages. The list of pages includes "Home", "Create Report in Embed View", "Custom Layout", "Embed Configuration Details", "Embed Single Visual", and "Embedding Basic interaction".

Home · Microsoft/PowerBI-JavaScript

GitHub, Inc. [US] | <https://github.com/Microsoft/PowerBI-JavaScript/wiki>

This repository Search Pull requests Issues Marketplace Explore

Microsoft / PowerBI-JavaScript Watch 102 Star 237 Fork

Code Issues 81 Pull requests 1 Projects 0 Wiki Insights

Home

ali-hamud edited this page 17 hours ago · 45 revisions

Welcome to the PowerBI-JavaScript wiki!

Embed:

- [Embedding Basics](#)
- [Embed Configuration Details](#)
- [Create Report In Embed View](#)
- [Service Configuration](#)

Reports:

- [Update Settings](#)

Pages 21

Find a Page...

- [Home](#)
- [Create Report in Embed View](#)
- [Custom Layout](#)
- [Embed Configuration Details](#)
- [Embed Single Visual](#)
- [Embedding Basic interaction](#)

Hello World with Power BI Embedding

- PBIJS library provides **powerbi** as top-level service object
 - You create configuration and then call **powerbi.embed** to embed a report
 - You must pass access token as part of the configuration

```
// data required for embedding Power BI report
var embedReportId = "f10c9de9-a325-4a43-af9f-0cf35cca2ab7";
var embedUrl = "https://app.powerbi.com/reportEmbed?reportId=f10c9de9-a325-4a43-af9f";
var accessToken = "H4sIAAAAAAEACWwtw6sCBZE_-WlrIR3K02A9x66gQzvVwe0_76tmbySW6pbdF7-Y";

// 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,
};

// 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);
```



Agenda

- Power BI Embedding Onboarding Experience
- Creating ASP.NET MVC Project for Power BI Embedding
- Retrieving Power BI Embedding Data
- Generating and Managing Embed Tokens
- Writing Client-side Code to Embed Resources

