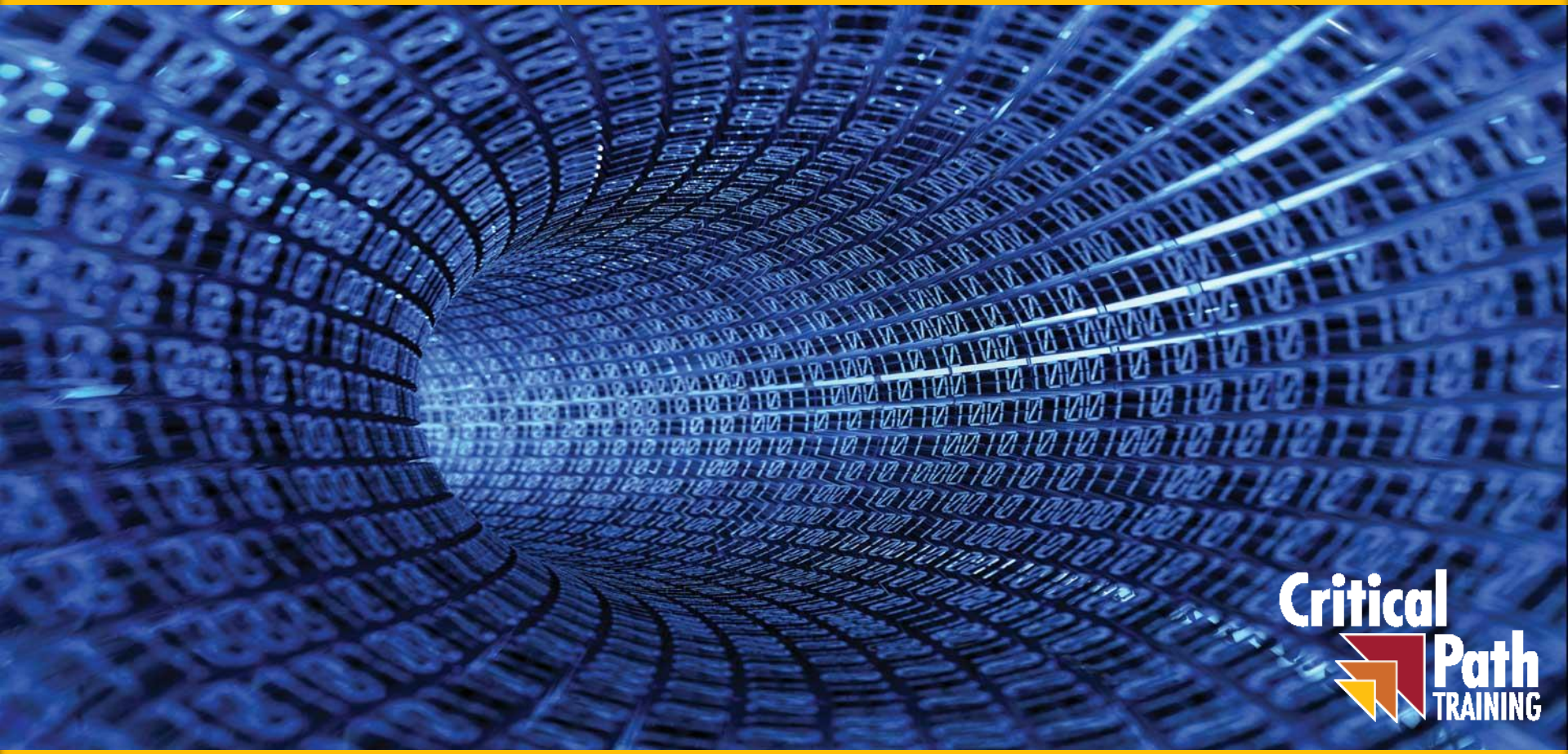
Embedding Power BI Reports with ASP.NET MVC

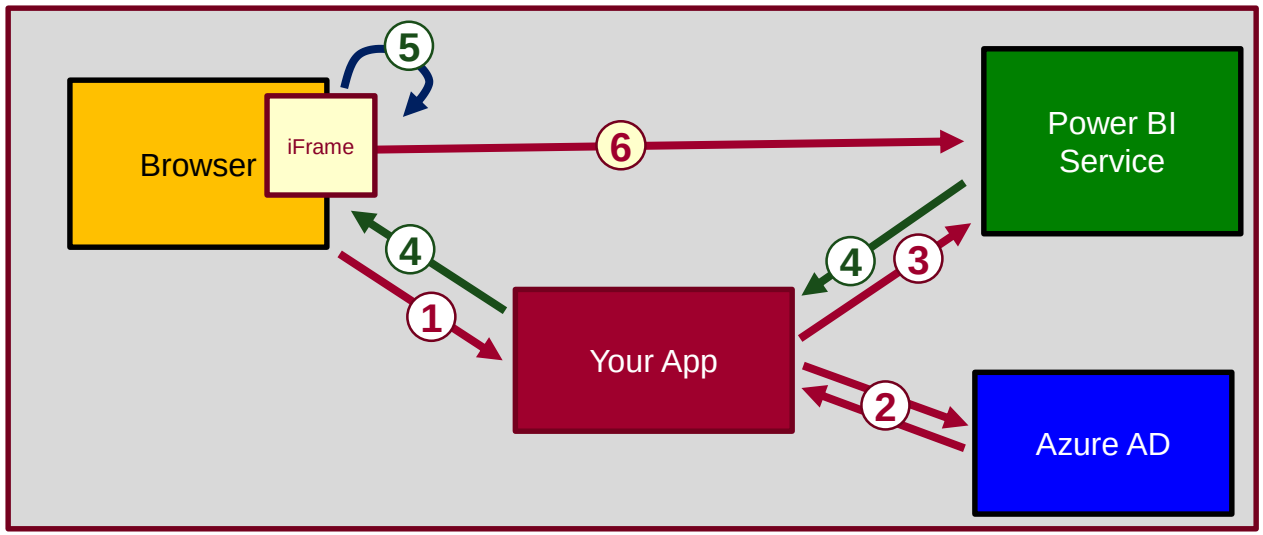


Agenda

- 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

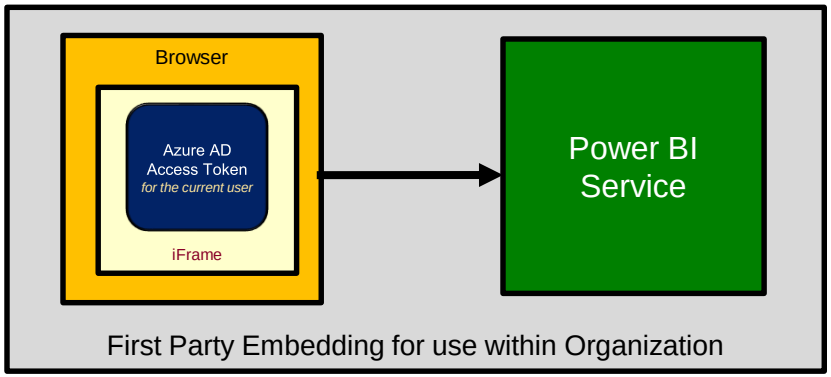
- User launches your app using a browser
- App authenticates with Azure Active Directory and obtains access token
- App uses access token to call to Power BI Service API
- App retrieves data for embedded resource and passes it to browser.
- Client-side code uses Power BI JavaScript API to create embedded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



First Party Embedding vs Third Party Embedding

First Party Embedding

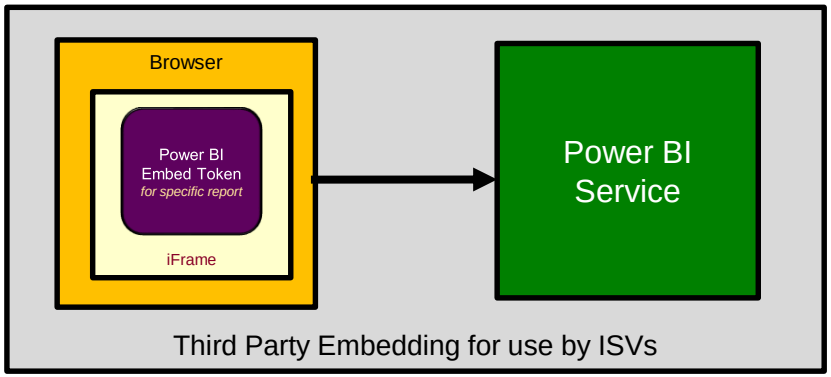
- Known as **User-Owns-Data** Model
- All users require a Power BI license
- Useful in corporate environments
- App authenticates as current user
- Your code runs with user's permissions
- User's access token passed to browser



First Party Embedding for use within Organization

Third Party Embedding

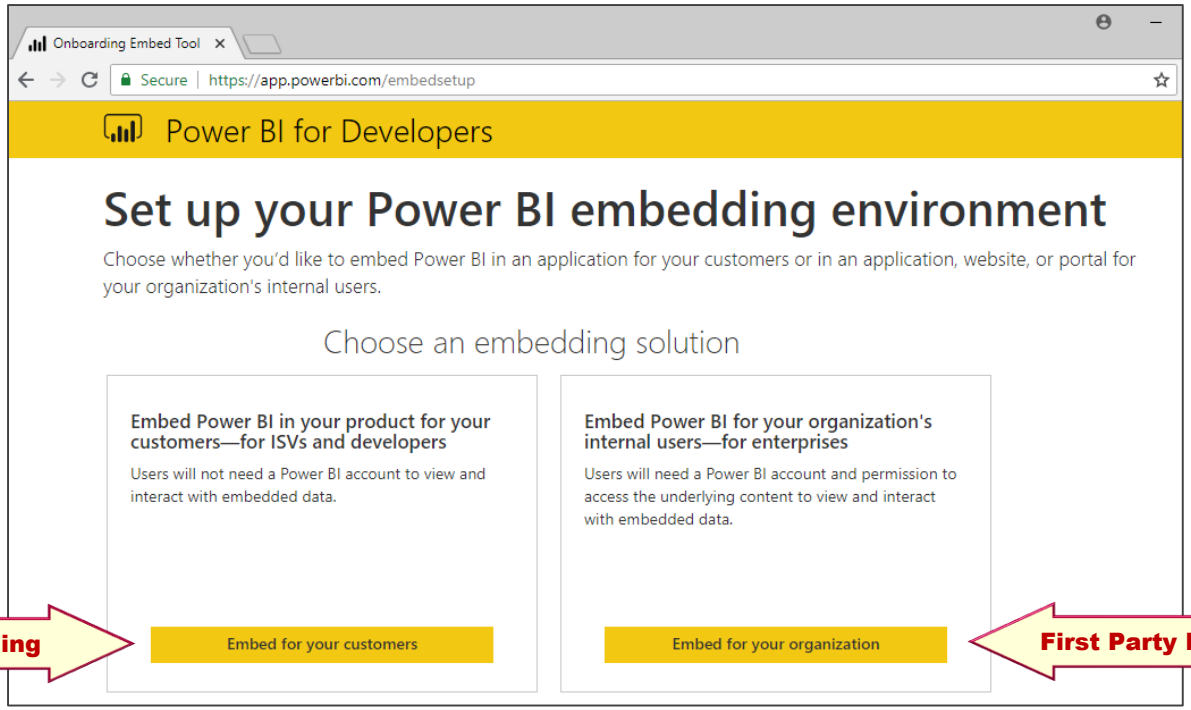
- Known as **App-Owns-Data** Model
- No users require Power BI license
- Useful for commercial applications
- App authenticates with master user account
- Your code runs with admin permissions
- Embed token passed to browser



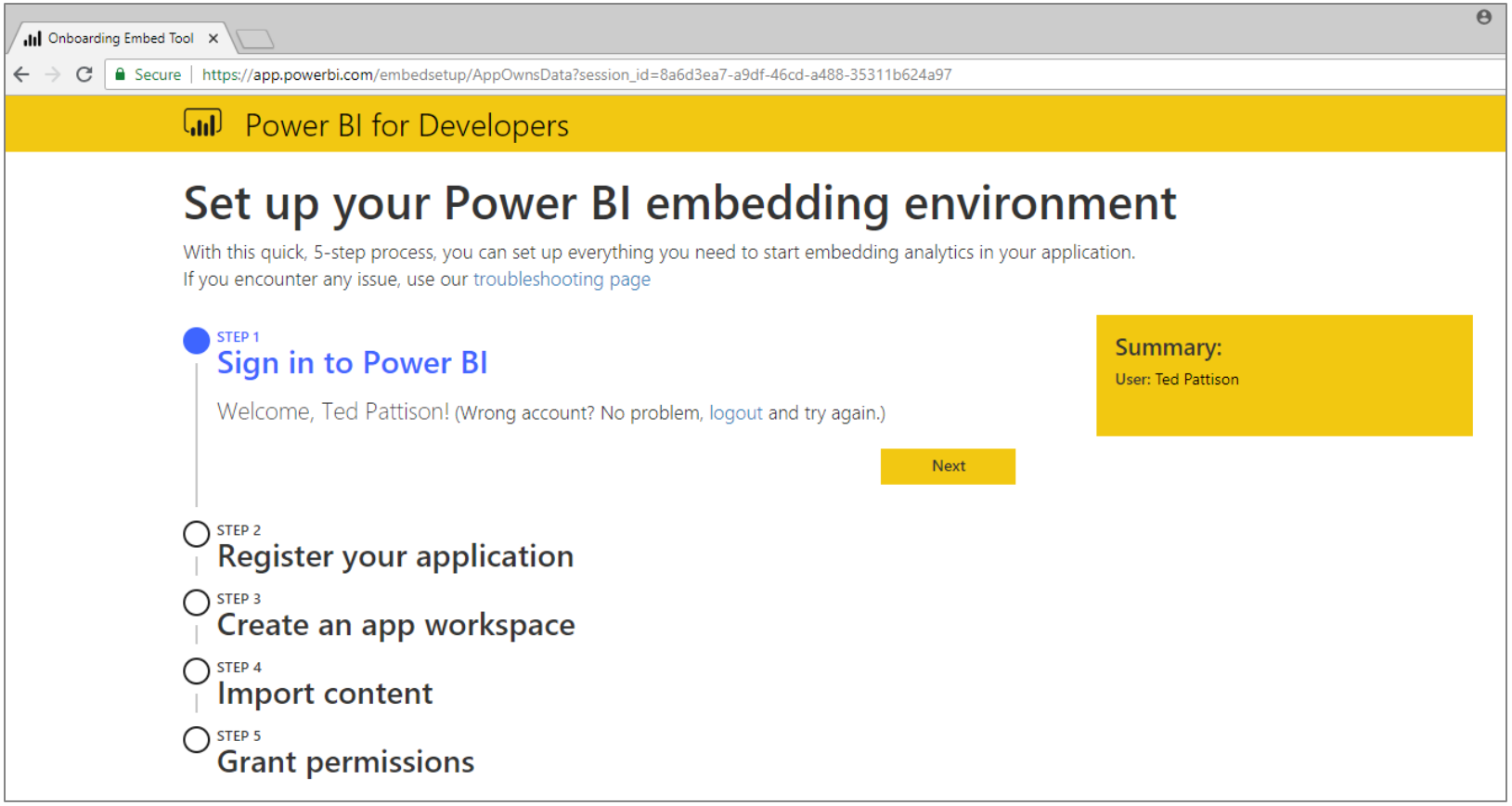
Third Party Embedding for use by ISVs

Power BI Embedding Onboarding Experience

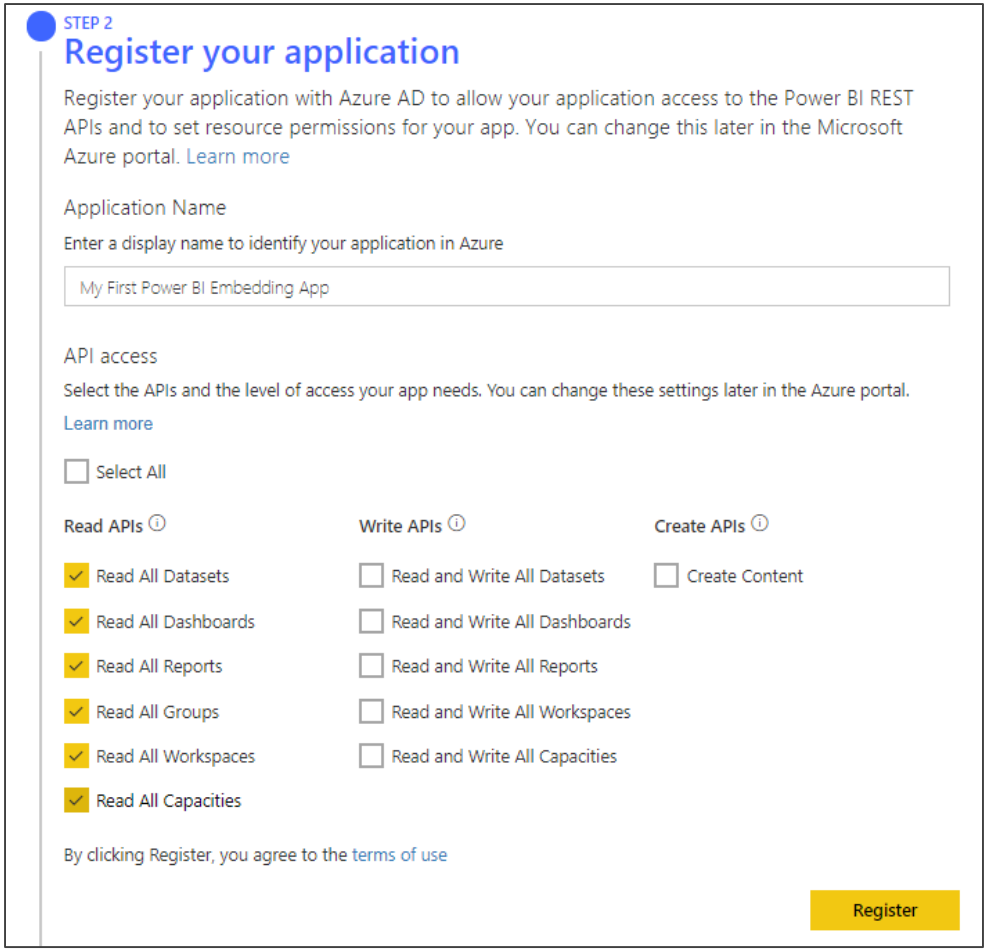
- https://app.powerbi.com/embedsetup



App-Owns-Data Experience – Step 1

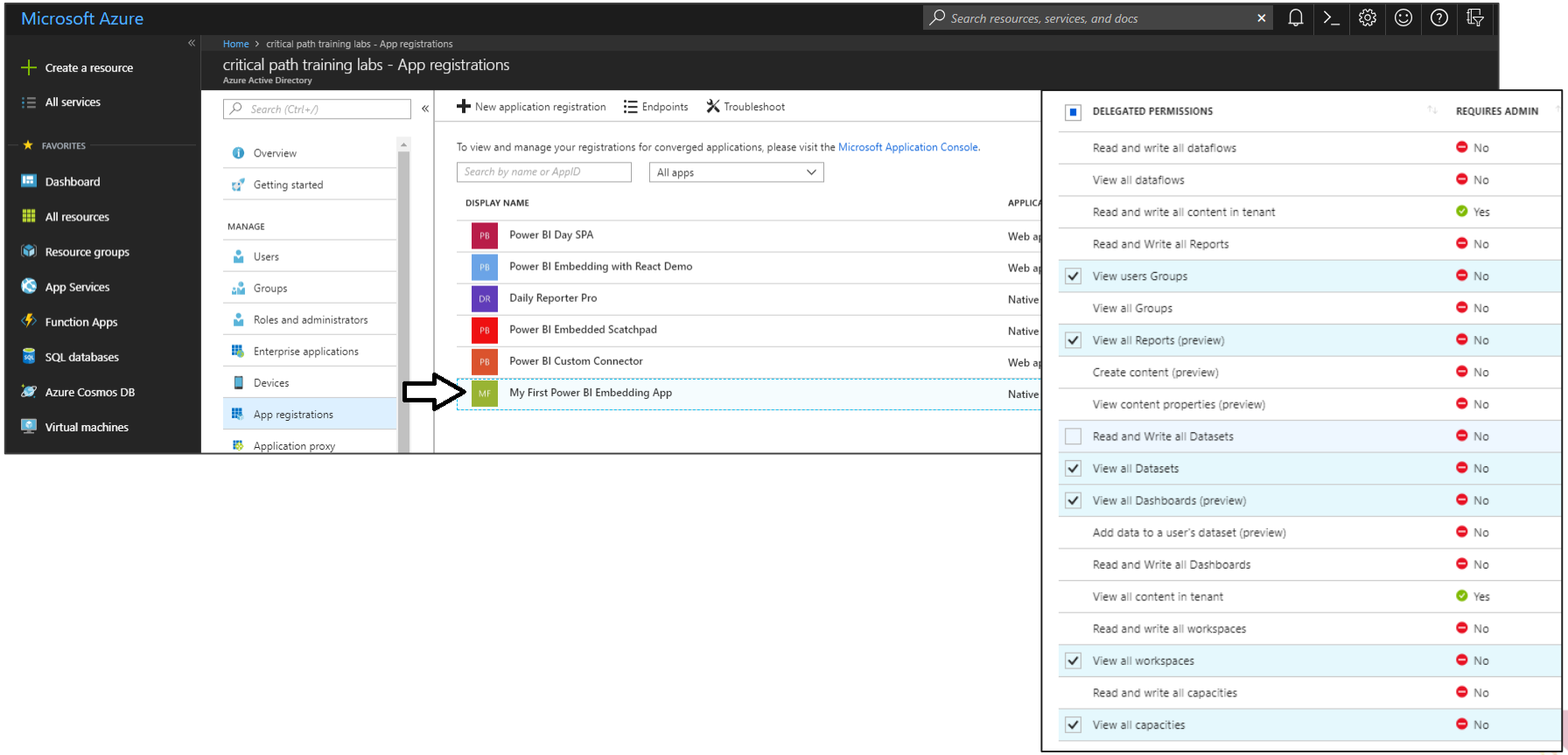


App-Owns-Data Experience – Step 2

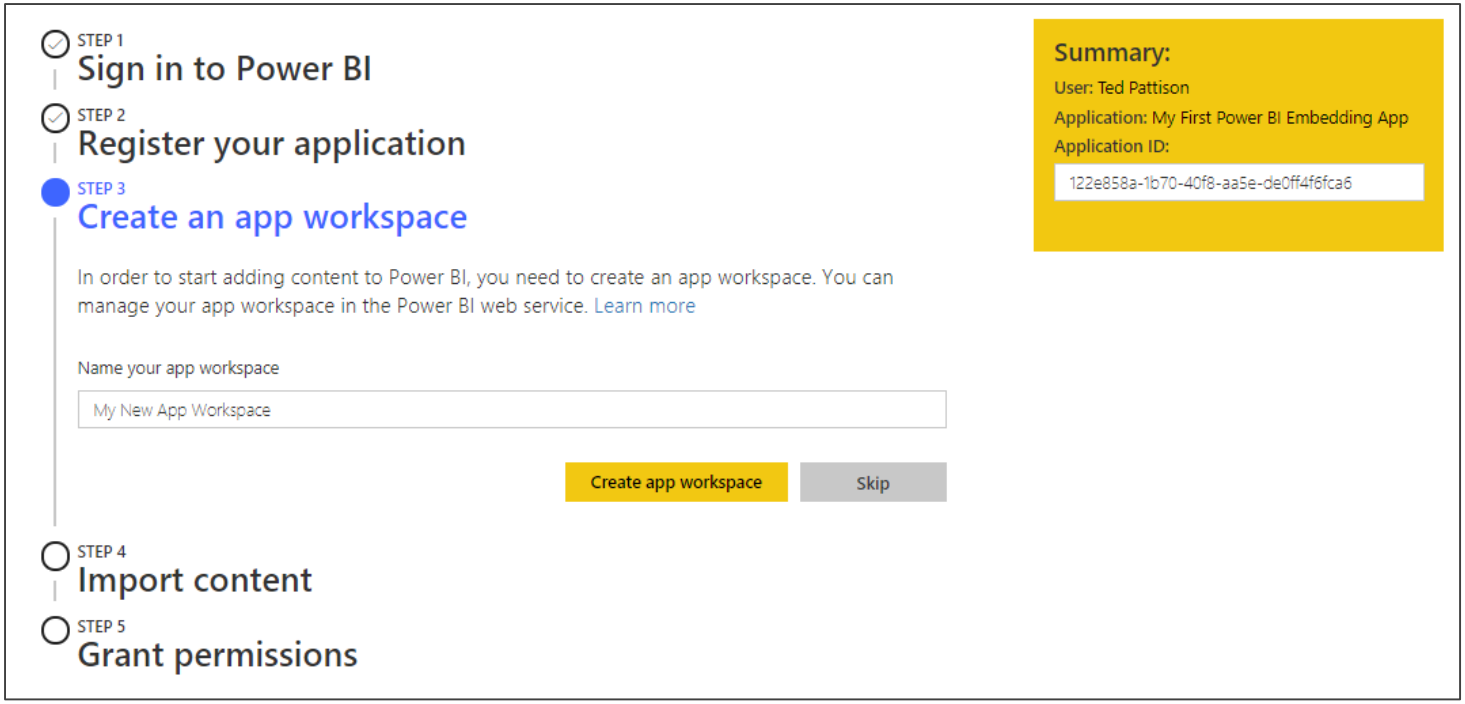


What Just Happened?

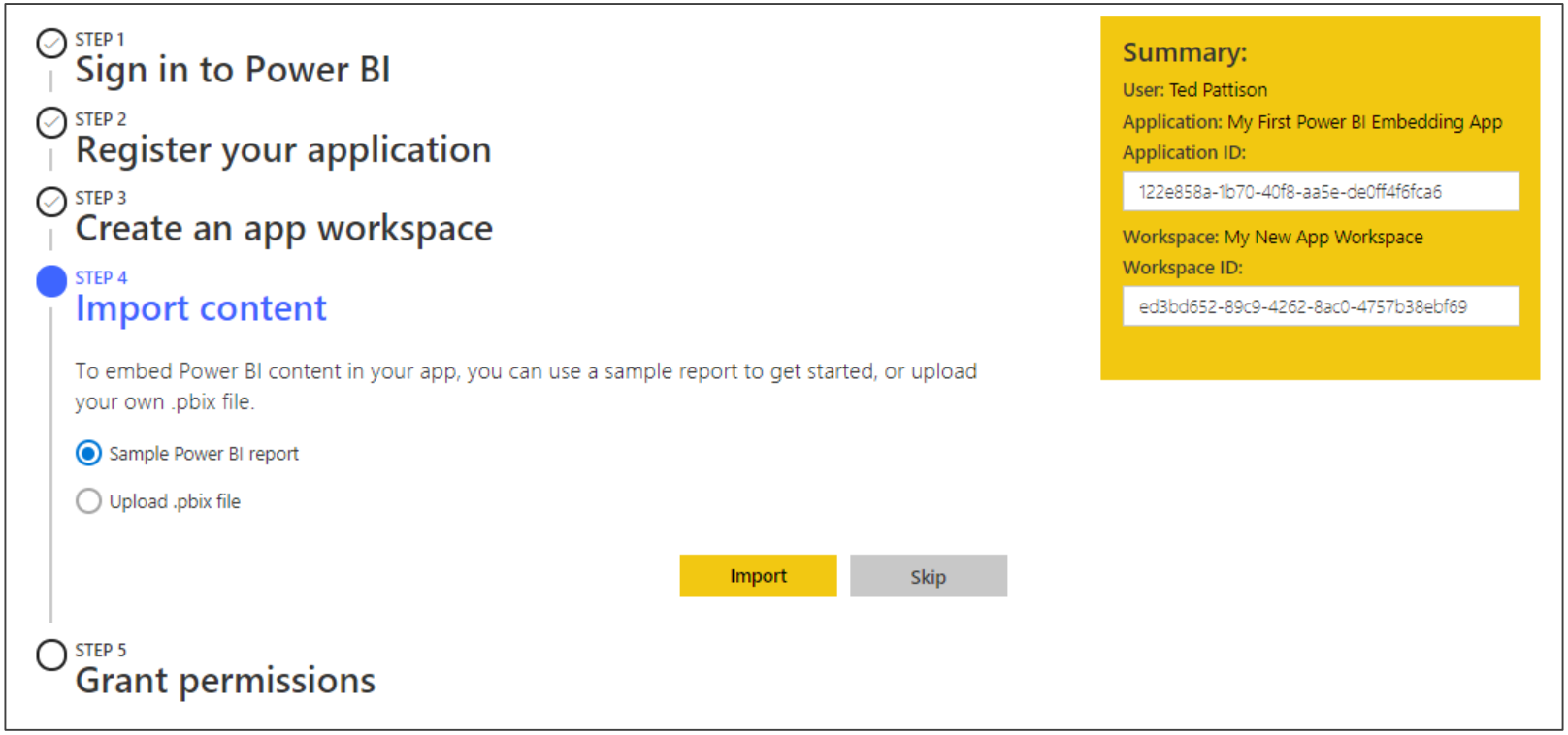
- The Onboarding Tool created a new Azure AD App
  - Configured as a Native app as opposed to a Web app / API



App-Owns-Data Experience – Step 3

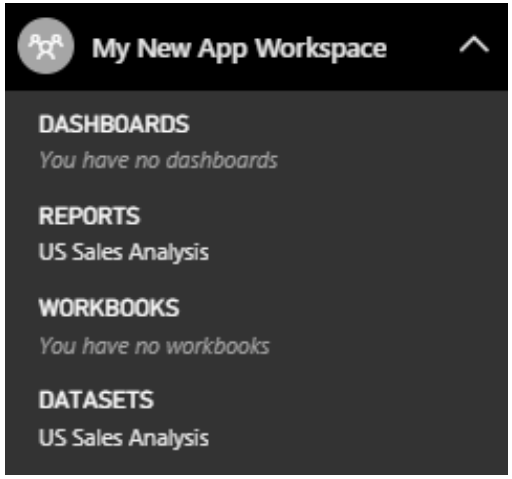


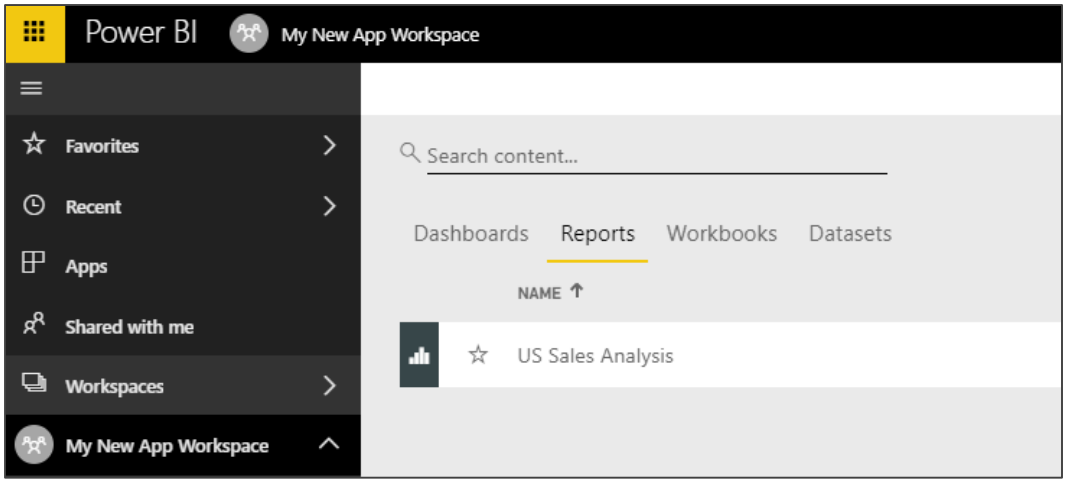
App-Owns-Data Experience – Step 4



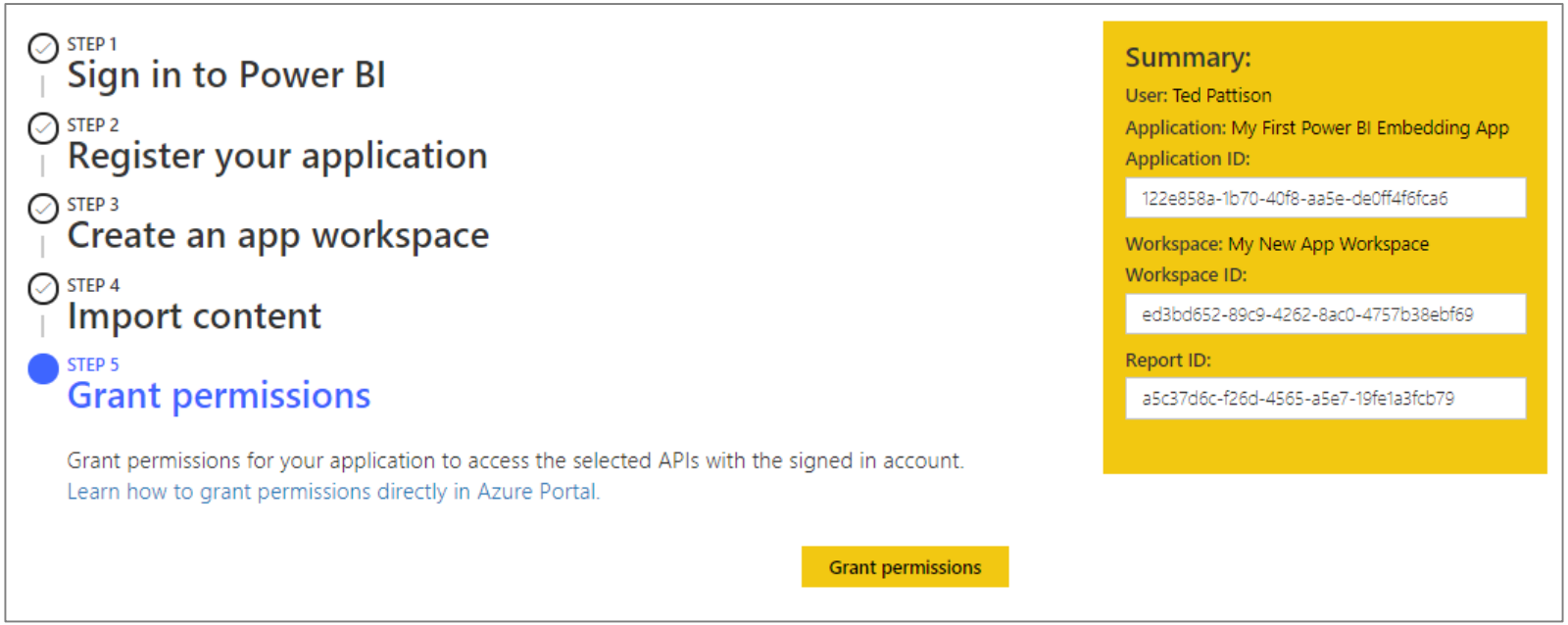
What Just Happened?

- 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



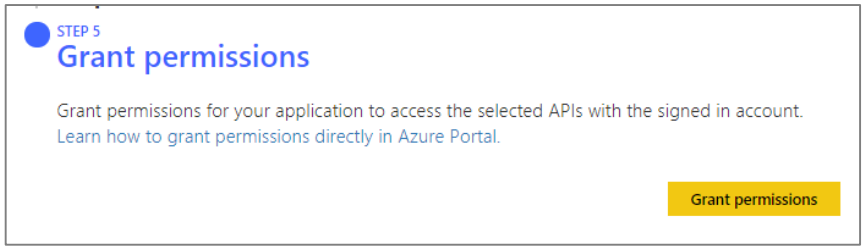


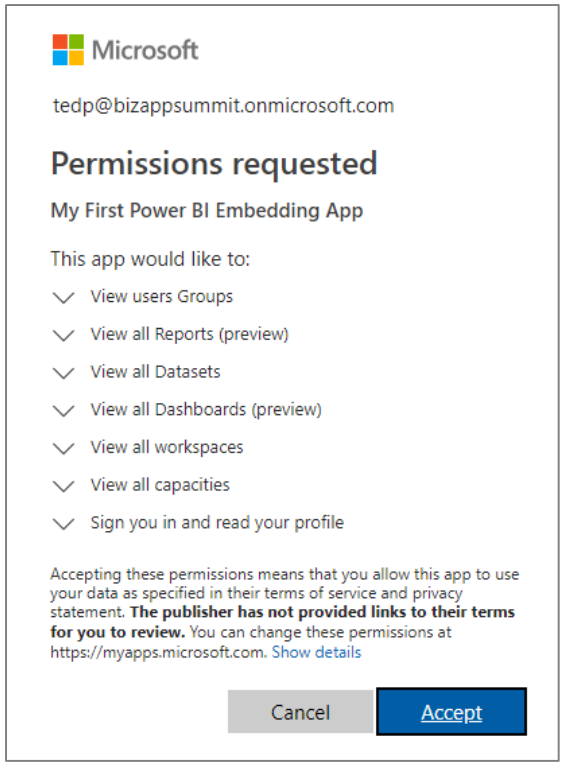
App-Owns-Data Experience – Step 5



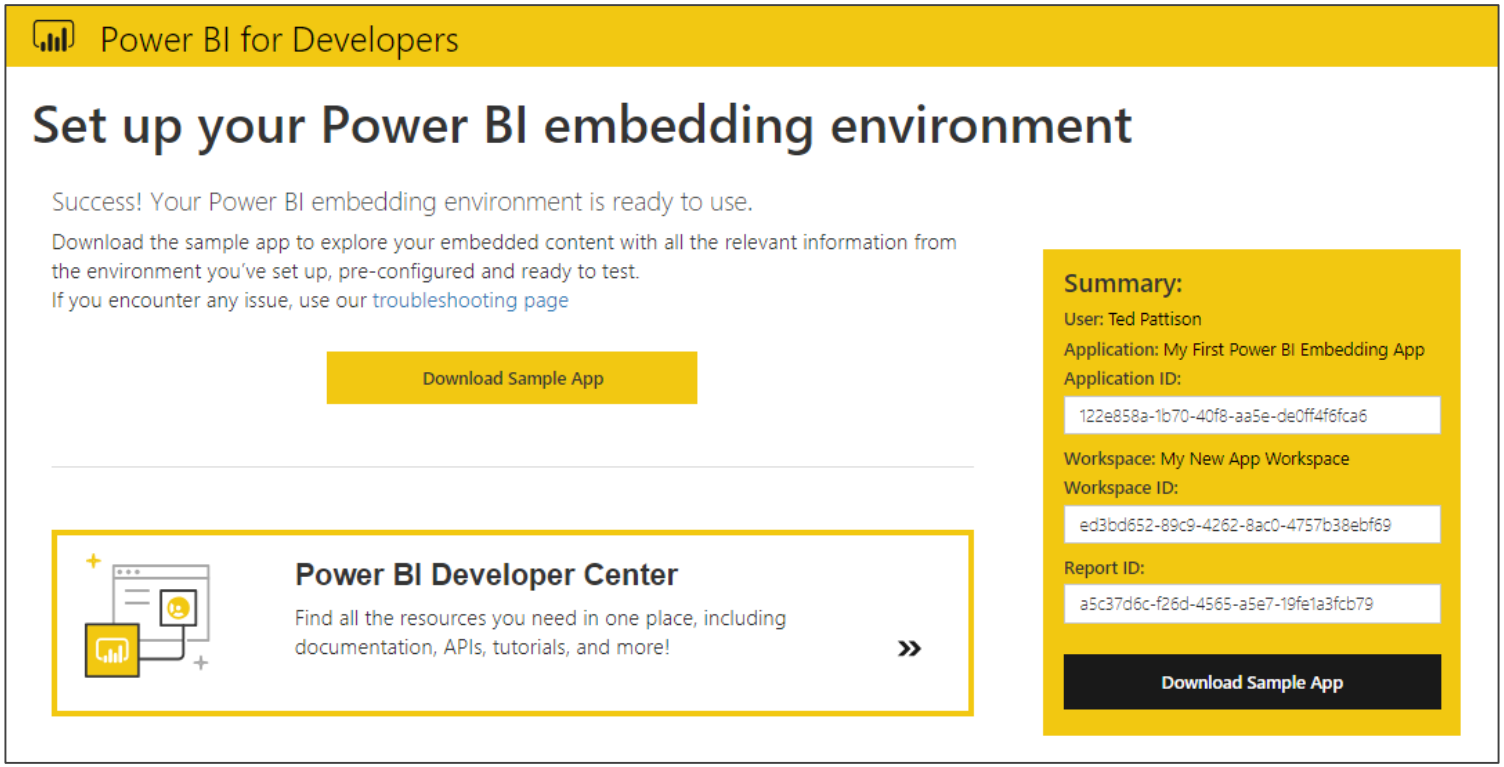
Granting Permissions to the App

- Users must consent to permissions requested by an app



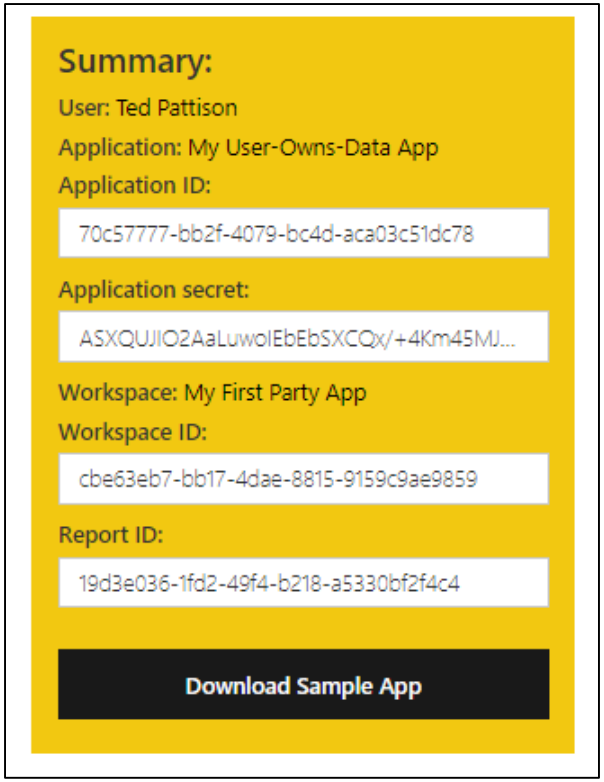


All Great Experiences Must Come to an End



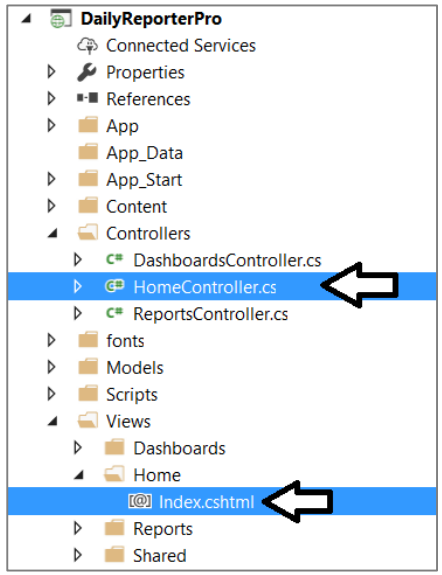
User-Owns-Data Onboarding Experience

- 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



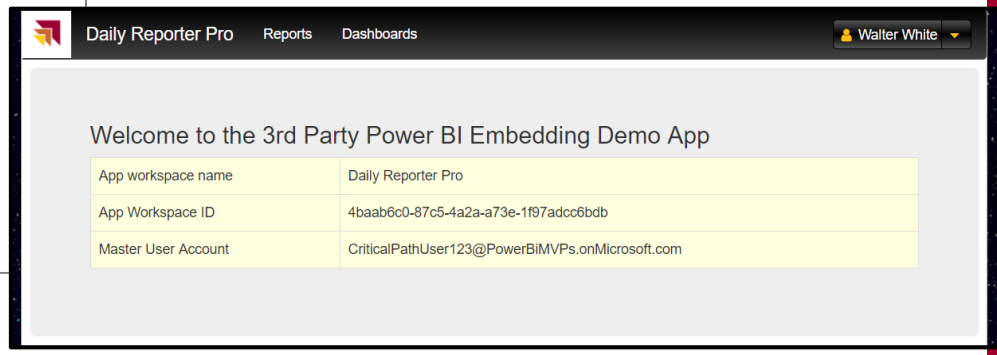
MVC Controllers and Views

```
public class HomeController : Controller {

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

}
```



Index.cshtml

```
@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

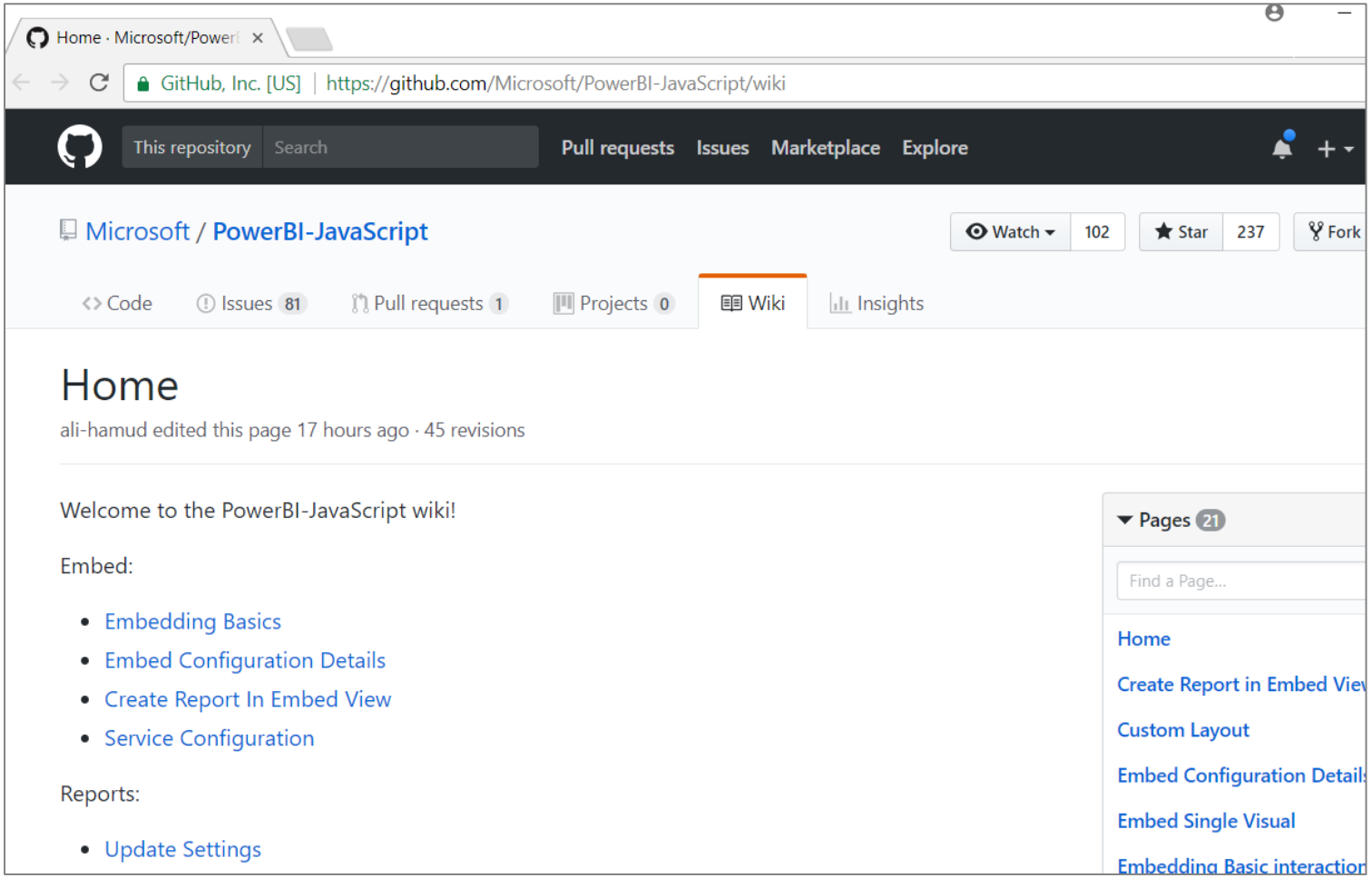
```
public static NewReportEmbeddingData GetNewReportEmbeddingData() {

  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



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 = "H4sIAAAAAAAAEACWWtw6sCBZE_-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