Aspecto destacado Nota

Module 8: Development with Power BI

Contents:

Module overview

The Power BI™ API is a REST-based API that developers use to access programmatically datasets, tables, and rows in Power BI. Using this API, you push data from an application into Power BI and integrate Power BI visualizations into an application. You can also add custom visuals to your applications and to Power BI dashboards and reports.

In this module, you will learn how to use the Power BI API to embed content in your applications and how to use custom visuals in your reports.

Objectives

After completing this module, you will be able to:

- Describe and use the Power BI API.
- List the steps for creating custom visualizations and import custom visuals into Power BI for use in Power BI reports.

Lesson 1: The Power BI API

At times, you may want to include Power BI reports, tiles, dashboards, and Q&A in other applications. For example, in your internal applications such as SharePoint Online, Microsoft Teams, and Dynamics 365, in custom applications within your organization, or in applications that you develop for external customers. Power BI supports all of these scenarios by using the Power BI API.

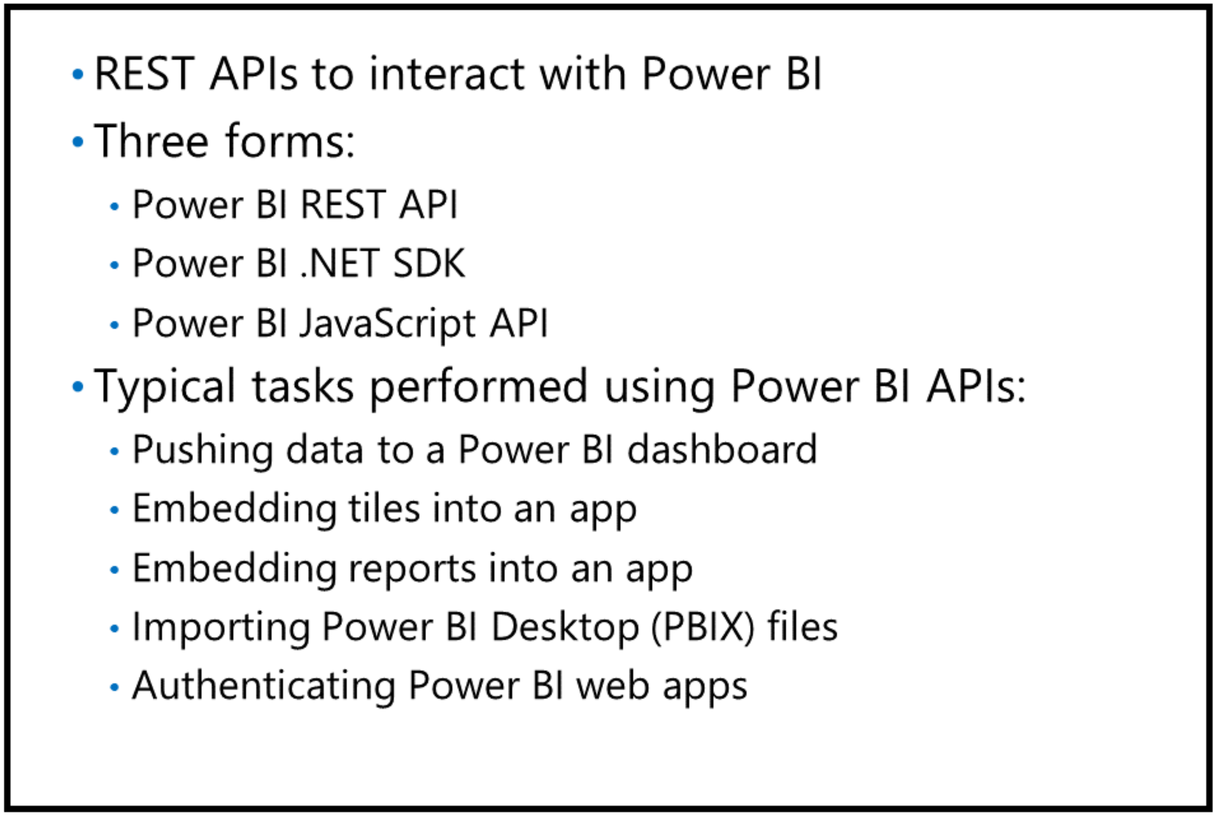
In this lesson, you will learn how to use the Power BI API in applications, to push data into Power BI, add data visualizations into applications, and customize visualizations. You will also see how to use the Power BI Embedded Playground to learn about the API and test your own applications.

Lesson objectives

After completing this lesson, you will be able to:

- Describe the key tasks that can be completed using the Power BI API.
- Describe how to embed Power BI content in your applications.
- Embed content for your organization using the Power BI API.
- Embed content for your customers using the Power BI API with Power BI Embedded.
- Use the Power BI Embedded Playground to learn about the Power BI API.

What is the Power BI API?



- REST APIs to interact with Power BI
- Three forms:
  - Power BI REST API
  - Power BI .NET SDK
  - Power BI JavaScript API
- Typical tasks performed using Power BI APIs:
  - Pushing data to a Power BI dashboard
  - Embedding tiles into an app
  - Embedding reports into an app
  - Importing Power BI Desktop (PBIX) files
  - Authenticating Power BI web apps

The Power BI API is a set of REST APIs from the Power BI Service platform that enables you to interact with Power BI from your applications. Developers use the API to push data into Power BI, display visuals from Power BI, and retrieve metadata about the objects in the service. This enables in-house developers and independent software vendors (ISVs) to integrate Power BI visuals, such as reports, tiles, dashboards, and Q&A content, into their own applications.

You use the Power BI API in three different forms:

- Power BI REST API – a REST API that you use from any programming language that supports REST calls.

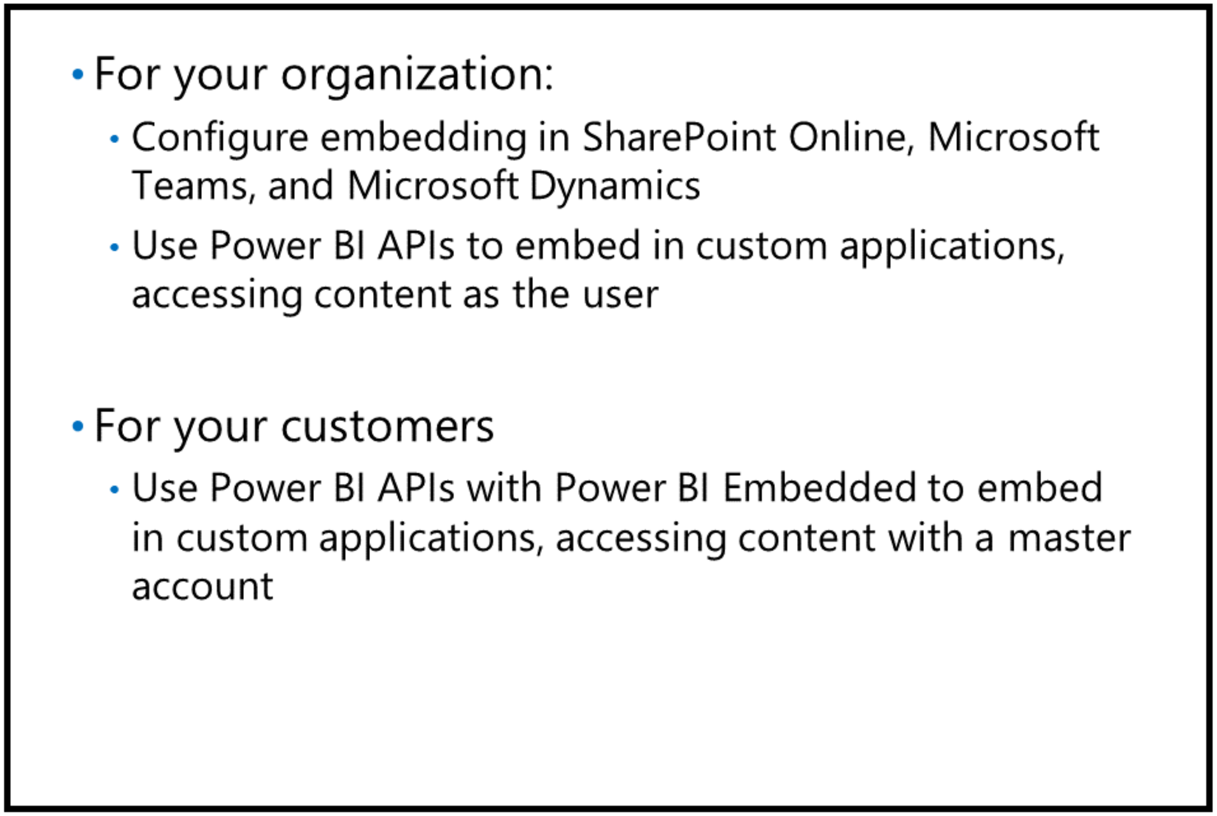
- Power BI .NET SDK – a wrapper to the Power BI API that you use from .NET languages to simplify your calls to the API.
- Power BI JavaScript API – a client-side library that you use to interact with your embedded content; for example, to filter a Power BI report displayed in your application.

In all of these scenarios, Power BI retains control of the authentication and authorization of the content, ensuring that unauthorized users can't access your data. You can even take advantage of the security features in Power BI, such as row level security.

Typical tasks performed using Power BI APIs include:

- Extending existing business workflows to push key data into a Power BI dashboard.
- Embedding tiles into an app.
- Embedding reports into an app.
- Importing Power BI Desktop (PBIX) files.
- Authenticating Power BI web apps.

Embedding Power BI content in your applications



- For your organization:
  - Configure embedding in SharePoint Online, Microsoft Teams, and Microsoft Dynamics
  - Use Power BI APIs to embed in custom applications, accessing content as the user
- For your customers
  - Use Power BI APIs with Power BI Embedded to embed in custom applications, accessing content with a master account

You can embed Power BI content into websites and applications designed for internal use and for external customers.

For your organization

You can easily embed Power BI content into various Microsoft systems, including SharePoint Online, Microsoft Teams, and Microsoft Dynamics 365. Embedding in these systems does not require any coding; you just create your Power BI content then configure the application to display it.

- **SharePoint Online**. Publish your report to your Power BI account, get the URL for that report, and then use the URL for a Power BI web part on your SharePoint Online page.
- **Microsoft Teams**. Add Power BI as a tab in your Microsoft Teams channel, and then select which reports to show in that channel.
- **Dynamics 365**. Enable Power BI visualization embedding in Dynamics 365, and then add a Power BI dashboard or tile to your Dynamics 365 dashboard.

You can also embed Power BI content into your own custom internal applications using the Power BI API. In this scenario, users utilize their existing Power BI accounts to access the content on the Power BI service they own or one that has been shared with them. They use the standard Power BI login procedure that returns an auth token that your application uses to access their data and reports as that user.

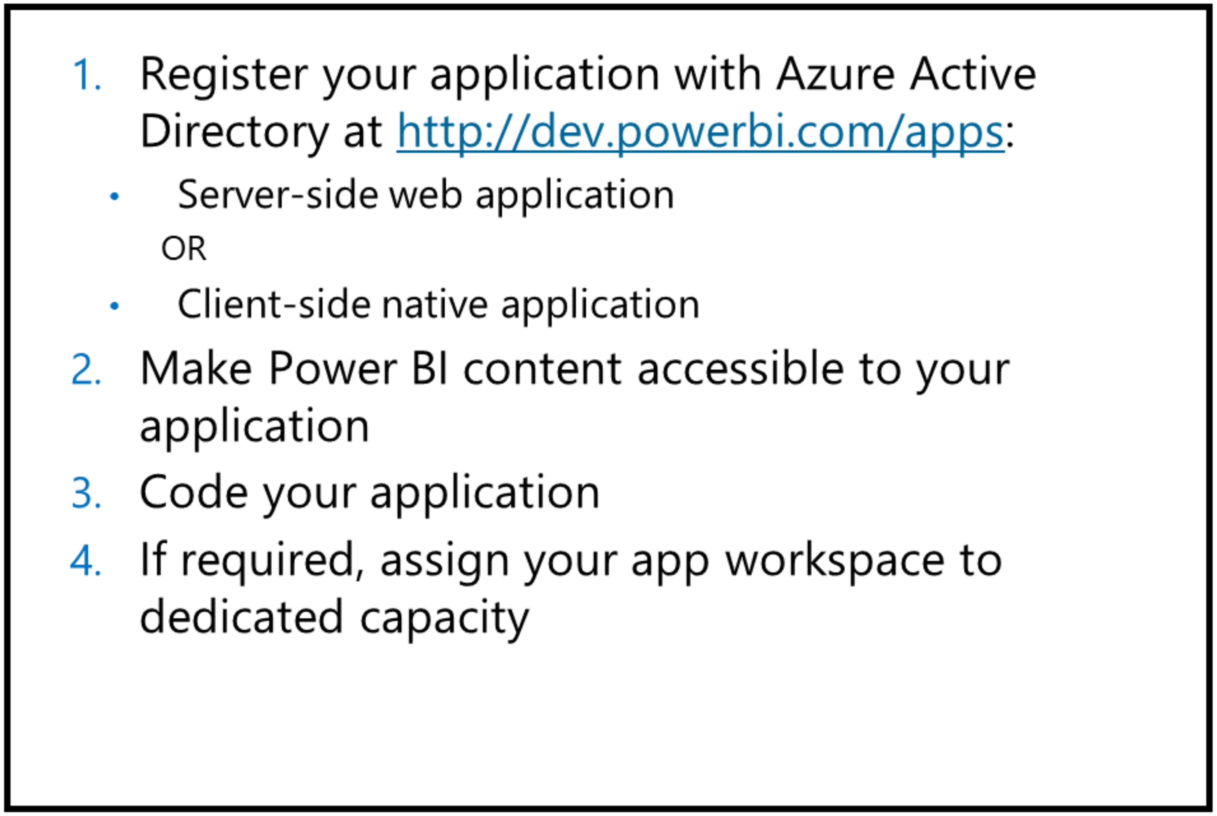
For your customers

When embedding content in applications for external customers, you're unlikely to know whether they have Power BI licenses. Even if they have licenses, they likely won't have access permissions on your content or data. In this scenario, you use the Power BI APIs in conjunction with Power BI Embedded to integrate reports, tiles, and dashboards seamlessly into your applications. Your users can work with these objects in your application without needing access to, or even knowing anything about, Power BI.

Your application accesses Power BI using a master account that has admin rights to the specific content that you want to embed. You can think of this account as a proxy account through which all users of your application will access Power BI content.

Power BI Embedded is a Microsoft Azure service that enables Power BI access using a capacity-based, hourly metered model. Using Azure capacity means that you can easily scale-up or scale-down resources and control the billing of your application.

Embedding for your organization



1. Register your application with Azure Active Directory at http://dev.powerbi.com/apps:
   - Server-side web application

   OR
   - Client-side native application
2. Make Power BI content accessible to your application
3. Code your application
4. If required, assign your app workspace to dedicated capacity

To embed Power BI content for your organization, you need to configure the application within Azure Active Directory to allow it to call the Power BI API. Then you set up the Power BI content to make it accessible to your application, and finally write code in your application to embed the content.

Register your application with Azure Active Directory

To register your application, perform the following steps:

1. In a web browser, go to **http://dev.powerbi.com/apps**.
2. Sign in using your Power BI credentials.
3. Specify the following, and then click **Register**:
    a. **Application Name** – to identify your application in Azure.
    b. **Application Type** – server-side web application or client-side native application.
    c. **Home Page URL** and **Redirect URL** – homepage and landing page after sign-in (for server-side web applications).
    d. **API access** – which API calls your application will use.

The **Success** dialog box returns an **Application ID** and, if you specified a server-side web application, an **Application secret** which you use in your application code.

Make Power BI content accessible to your application

To enable embedding of content, place your reports, dashboards, and tiles in an app workspace, and then publish that workspace to Power BI.

Code your application

Use the Power BI REST API, the Power BI .NET SDK, or the Power BI JavaScript API to write code to embed content in your application:

1. Use the **Application ID** and, if writing a server-side web application, the **Application secret** from the registration process to authenticate your user and create an auth token for Power BI.
2. Use Power BI API calls to perform the tasks your application requires. For example, you could use the auth token to obtain an embed URL for a Power BI report, and then use that URL to embed the report in your application.

For code examples showing how to embed Power BI content for your organization, see the *User Owns Data sample app* on GitHub.

User Owns Data sample app

https://aka.ms/AA55hb5

If required, assign your app workspace to dedicated capacity

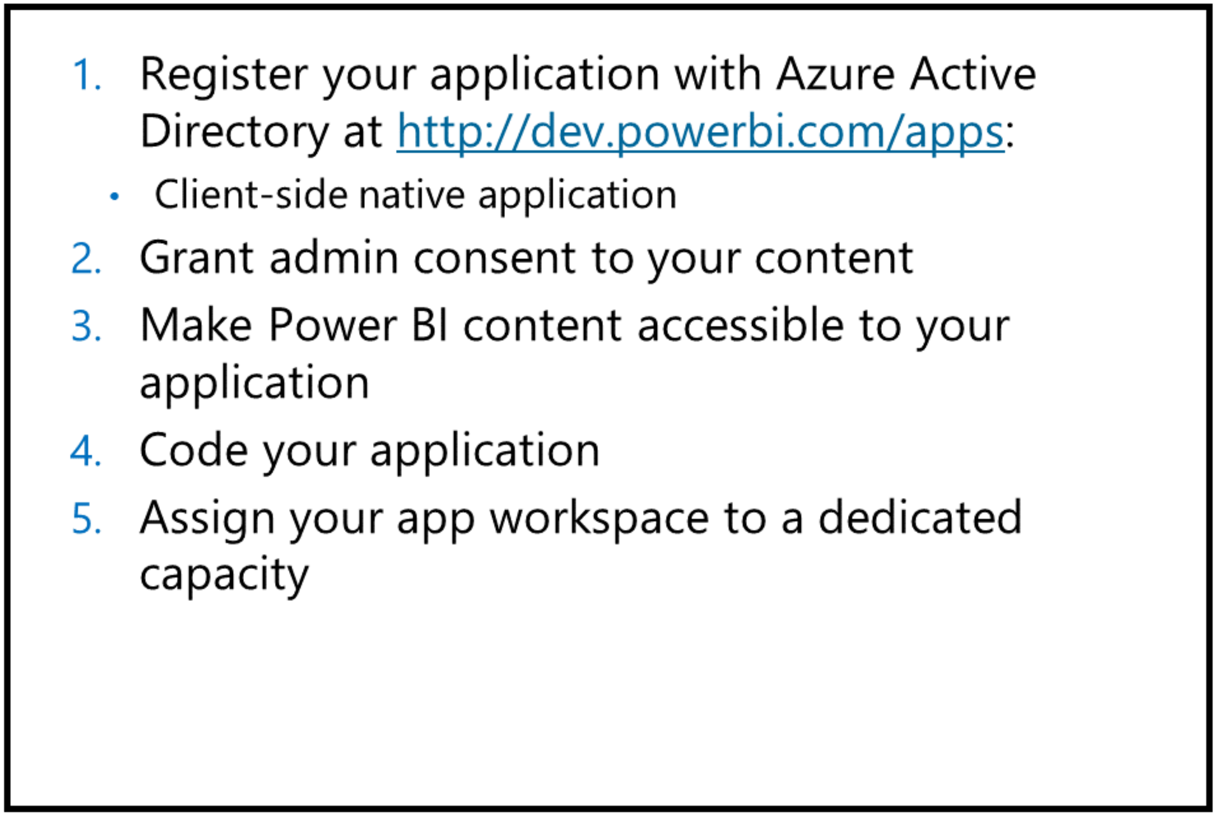
If all of your internal users have Power BI Pro licenses, they will be able to use your application as is. However, if you want free users to be able to access it, you need to purchase capacity for the app workspace that contains your Power BI content and grant those free users permission to access the app workspace.

For more information about embedding Power BI content for your organization, see *Tutorial:Embed Power BI content into an application for your organization* in the Power BI documentation.

Tutorial: Embed Power BI content into an application for your organization

https://aka.ms/AA553vu

Embedding for your customers



1. Register your application with Azure Active Directory at http://dev.powerbi.com/apps:
   - Client-side native application
2. Grant admin consent to your content
3. Make Power BI content accessible to your application
4. Code your application
5. Assign your app workspace to a dedicated capacity

Embedding Power BI content for your customers is a similar process to embedding it for your organization. However, it does contain additional steps and requirements to ensure that the application works seamlessly with your content, without the user interacting with the Power BI user interface.

Register your application with Azure Active Directory

To register your application, perform the following steps:

1. In a web browser, go to **http://dev.powerbi.com/apps**.

2. Sign in using your Power BI credentials.

3. Specify the following, and then click **Register**:

    a. **Application name** – to identify your application in Azure

    b. **Application type** – choose client-side native application for Power BI Embedded applications that work with non-interactive logins

    c. **API access** – which API calls your application will use

The **Success** dialog box returns an **Application ID** that you use in your application code.

Grant admin consent to your content

To enable your application to make Power BI API calls without any user interaction, you need to grant consent for the master account user to make the calls in your Azure portal.

1. Sign in to your Azure portal.

2. On the **App registrations** blade, click your application name.

3. On your application blade, click **API permissions**, click **Grant admin consent for <your user name>**, and then in the message box, click **Yes**.

4. Wait for the grant process to complete.

Make Power BI content accessible to your application

To enable embedding of content, place your reports, dashboards, and tiles in an app workspace, and then publish that workspace to Power BI.

Code your application

Use the Power BI REST API, the Power BI .NET SDK, or the Power BI JavaScript API to write code to embed your content in your application:

1. Use the **Application ID** from the registration process to authenticate your master account and create an auth token for Power BI.
2. Use Power BI API calls to perform the tasks your application requires. For example, you could use the auth token to obtain an embed URL for a Power BI report, use that URL to create an embed token for the report, and then use the embed token to embed the report in your application.

For code examples showing how to embed Power BI content for your customers, see the *App Owns Data sample app* on GitHub.

App Owns Data sample app

https://aka.ms/AA55hb5

Assign your app workspace to a dedicated capacity

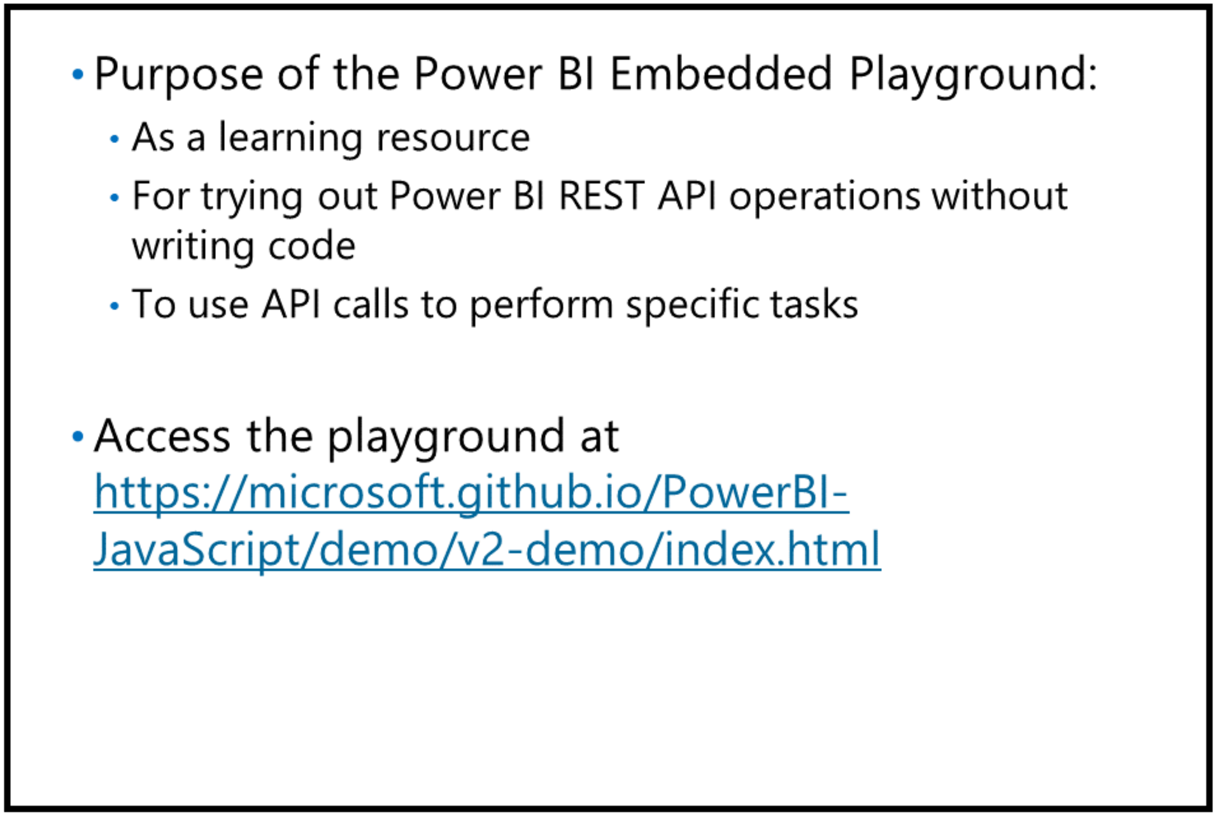
When using Power BI Embedded for your customers, your app workspace must be assigned to a dedicated capacity.

For more information about embedding Power BI content for your customers, see *Tutorial: Embed Power BI content into an application for your customers* in the Power BI documentation.

Tutorial: Embed Power BI content into an application for your customers

https://aka.ms/AA553vs

The Power BI Embedded Playground



- Purpose of the Power BI Embedded Playground:
  - As a learning resource
  - For trying out Power BI REST API operations without writing code
  - To use API calls to perform specific tasks
- Access the playground at https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html

The Power BI Embedded Playground is a good starting place for learning about the Power BI API, for trying out Power BI REST API operations without needing to write any code, and for using API calls to perform specific tasks against your data.

Browse to the site at **https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html**. On the **Samples** page, you can access code samples that show how to embed the following items in your applications:

- Report
- Report Visual

- Q&A
- Dashboard
- Tile

By default, you can review and run the sample code against the sample data and copy the sample code for your own use. You can also change the settings to run the code against your own Power BI content directly in the playground.

On the **Showcase** page, you can review and learn about new features in Power BI Embedded and the **Documentation** page contains links to useful content, in addition to videos about Power BI Embedded.

Demonstration: Using the Power BI Embedded Playground

In this demonstration, you will see how to use the Power BI Embedded Playground to learn about the Power BI API.

Lesson 2: Custom visuals

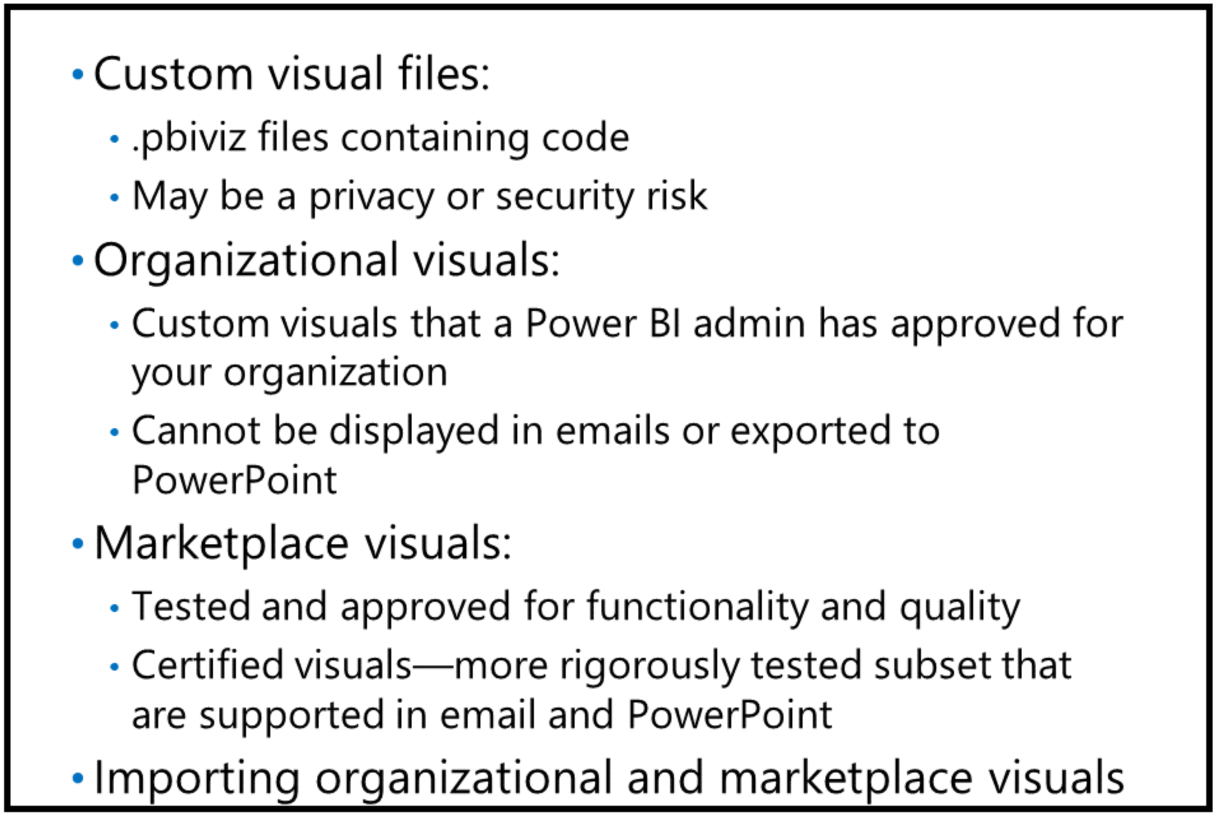
In this lesson, you will learn how to add custom visuals to reports, dashboards, and workspaces and how to use the Microsoft Power BI visuals project to create your own custom visualizations. You will also learn how to share custom visuals through the Power BI visuals gallery.

Lesson objectives

After completing this lesson, you will be able to:

- Import custom visuals from the Power BI visuals gallery into Power BI and use them in Power BI reports.
- Describe the process used for creating custom visualizations for use with Power BI applications and dashboards.

Using custom visuals



- Custom visual files:
  - .pbiviz files containing code
  - May be a privacy or security risk
- Organizational visuals:
  - Custom visuals that a Power BI admin has approved for your organization
  - Cannot be displayed in emails or exported to PowerPoint
- Marketplace visuals:
  - Tested and approved for functionality and quality
  - Certified visuals—more rigorously tested subset that are supported in email and PowerPoint
- Importing organizational and marketplace visuals

Power BI Desktop and the Power BI service both provide a whole range of visualizations that you use to display information in your reports. There are, however, additional visuals that you can use called *custom visuals*.

There are three types of custom visual that you can use in your reports:

1. **Custom visual files**. These are .pbiviz files that you import into a Power BI report to render your data. Any developer can create a .pbiviz file and, as such, the code they contain might present a privacy or security risk. Only import custom visual files from authors and sources that you know you can trust.

2. **Organizational visuals**. These are custom visuals that a Power BI admin has approved for use in your organization. They cannot be rendered in emails if a user subscribes to the report or exports it to PowerPoint.

3. **Marketplace visuals**. These are custom visuals created by Microsoft and members of the Power BI community that Microsoft has tested and approved for functionality and quality. Power BI certified visuals are a subset of the marketplace visuals that have been tested more rigorously for quality and are supported in email and PowerPoint scenarios.

Importing organizational and marketplace visuals

The steps to import organizational or marketplace visuals are similar; you just need to ensure that you're viewing the relevant list of available visuals in the **Power BI Visuals** dialog box. You can only import custom visuals into one report at a time.

To import a custom visual:

1. In Power BI Desktop, in the **VISUALIZATIONS** pane, click the ellipsis, and then click **Import from marketplace**.

2. If the visual you want is an organizational visual, in the **Power BI Visuals dialog** box, click **MY ORGANIZATION**.

3. Either browse the categories or use the **Search** box to find the visual that you want to use.

4. Click the visual name to review a description of the visual.

5. Click **Add** to import the visual to Power BI.

6. In the **Import custom visual** message box, click **OK**.

You can then use the imported visual to present your data in the same way that you use other visualizations in Power BI.

You can also download and import custom visuals from Microsoft AppSource at **https://appsource.microsoft.com/**. This method enables you to download a sample report showing how to use the visual.

1. In Internet Explorer, go to **https://appsource.microsoft.com/**.

2. On the **Apps** menu, click **Apps**.

3. In the **Products** list, click **Power BI visuals**.

4. Either browse the categories or use the **Search Microsoft AppSource** box to find the visual that you want to use.

5. Click the visual name to see a description of the visual and reviews from other users.

6. Click **GET IT NOW**, and in the **Sign in to Microsoft AppSource** dialog box, enter your email address, and then click **Sign in**.

7. If prompted, review the terms of use and privacy policy, and if you agree, click **Continue**.

8. On the **Install** page, you can either download the visual as a .pbiviz file or download a sample report that uses the visual as a .pbix file.

9. In Power BI Desktop, in the **VISUALIZATIONS** pane, click the ellipsis, and then click **Import from file**.

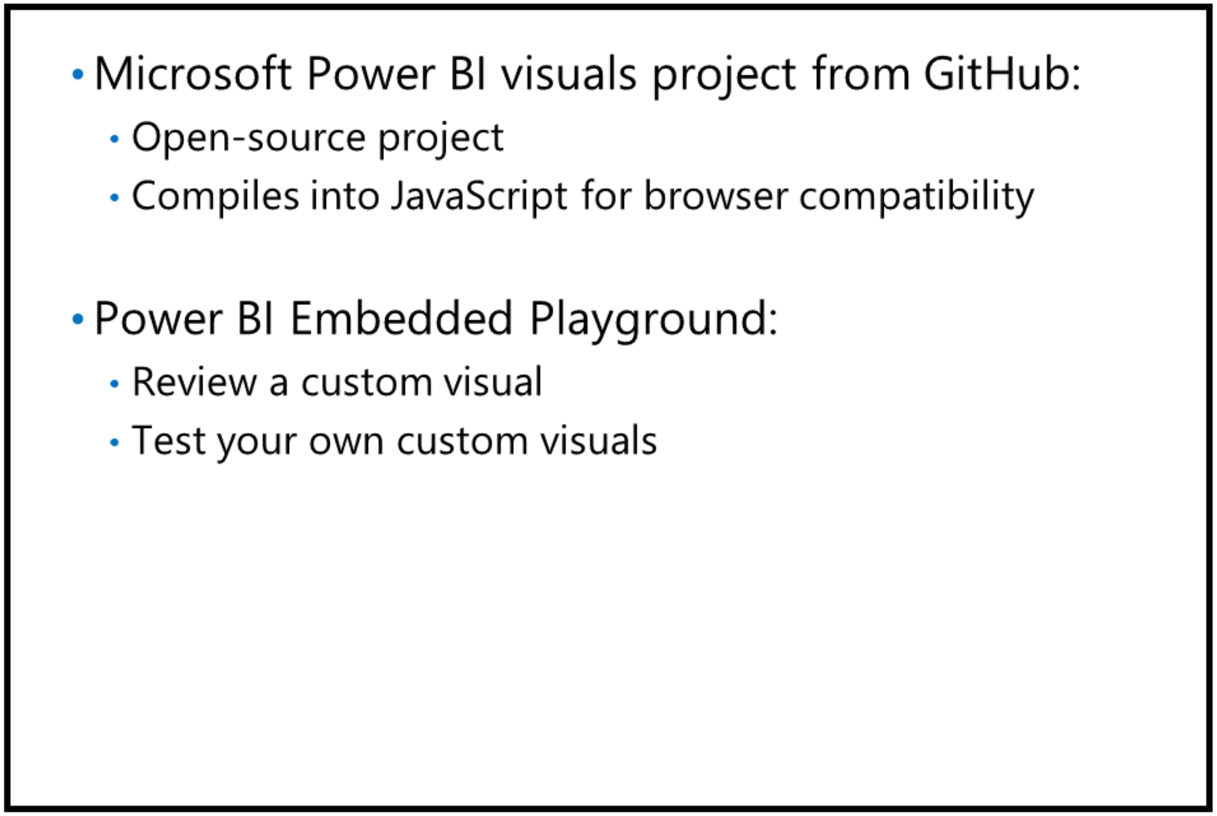
10. In the **Caution: Import custom visual** dialog box click **Import.**

11. Browse to the location of your downloaded .pbiviz file, select the file, and then click **Open.**

12. In the **Import custom visual** message box, click **OK**.

13. Optionally, open the downloaded report file, .pbix file, to review how to use the custom visual in a report.

Creating custom visuals



- Microsoft Power BI visuals project from GitHub:
  - Open-source project
  - Compiles into JavaScript for browser compatibility
- Power BI Embedded Playground:
  - Review a custom visual
  - Test your own custom visuals

If you need to render data that is not facilitated using either the in-built visualizations or those from the organizational and marketplace lists, you can create your own custom visuals. You either extend the capabilities of the in-built visuals or develop your own from scratch.

The code for many Power BI visuals is available in the Microsoft Power BI visuals project from GitHub. This open-source project consists of visualization code, tooling, and a test suite. The project includes more than 20 types of visualization, the framework that’s needed to run the visuals, and a testing infrastructure. The framework provides the required interface for integrating with the selecting controls, the filtering controls, and other user interface controls in Power BI. Because the code is written in TypeScript, it makes the visuals straightforward to build and debug. The visuals are built by using D3 (although you can use WebGL, Canvas, or SVG), and they compile into JavaScript and are compatible with modern browsers. This combination of technologies enables you to build your own custom visuals quickly.

You use the Power BI Embedded Playground at **https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html** to learn about coding custom visuals, in addition to testing how they look by using the Desktop or Phone view. You can also view and test code that interacts with the visual, such as filtering code and menu extensions.

For more information about creating custom visuals, see *Tutorial: Developing a Power BI custom visual* in the Power BI documentation:

Tutorial: Developing a Power BI custom visual

https://aka.ms/Y7zusi

Demonstration: Importing and using a custom visual

In this demonstration, you will see how to:

- Import a custom visualization into the Power BI Desktop.
- Use a custom visualization in a report.

Lab: Using marketplace visualizations

Scenario

Adventure Works employees are using Power BI to gain insights into sales patterns and trends, and for most purposes, the standard visualizations are all that is required. However, for some data there might be scope for using other types of visualization.

As a BI professional, you are asked to explore the use of custom visualizations, so that Adventure Works employees extract maximum value from sales datasets.

Objectives

After completing this lab, you will be able to use a custom visualization with Power BI data.

> **Note:** Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual − but you can access them on GitHub.

Lab setup

Estimated time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_08.md.

Exercise 1: Use a custom visualization

Scenario

As a data analyst for AdventureWorks, you are investigating the types of visualizations that can be used with sales data. For some data, it’s suggested that custom visualization might help AdventureWorks employees make better business decisions.

In this exercise, you will apply the Sunburst custom visualization to an existing report and compare this visualization with the standard visual that was previously in use.

The main tasks for this exercise are as follows:

1. Prepare the lab environment.
2. Using custom visuals.

Result: At the end of this exercise, the Sunburst custom visualization will be used in a Power BI report.

Review question(s)

Check your knowledge

Discovery

Do you think that the Sunburst visualization provides additional insights into the Sales Order data, compared with the clustered column chart that was originally used?

Show solution Reset

Check your knowledge

Discovery

From your own experience, are there any other custom visuals from the Power BI visuals gallery that would add value to the Sales Order data?

Show solution Reset

Module review and takeaways

In this module, you learned about the Power BI API and how developers use it to create applications. You also learned about the Power BI Embedded Playground and how to use custom visuals in Power BI reports.

Review question(s)

Check your knowledge

Discovery

Discuss the potential of the Power BI Developer API for your own organization. Are there any particular Power BI-based applications that you already use, or would like to see developed?

Show solution Reset