

Power-Bi reports in angular apps using iframe.

This is the most basic way of displaying power-bi reports in your apps and so allows minimal setup/configuration.

Here is a step-by-step guide to display a Power BI report in an iframe in an Angular application having angular version <13:

Step 1: Publish the Power BI report to the Power BI service

Before you can embed a Power BI report in your Angular application, you need to publish the report to the Power BI service. You can do this by following these steps:

1. Open the Power BI Desktop application and create your report.
2. Click on the "Publish" button in the "Home" tab of the Power BI Desktop application.
3. Select the workspace and report name, and then click on "Publish" to publish the report to the Power BI service.

Step 2: Obtain the embed URL for the Power BI report

Once the report is published to the Power BI service, you can obtain the embed URL for the report by following these steps:

1. Open the Power BI service in your web browser and navigate to the report.
2. Click on the "File" menu in the top left corner of the screen and select "Embed report".
3. In the "Embed report" pane, click on the "Create" button to create an embed token for the report.
4. Copy the entire embed URL from the "Embed URL" field.

Step 3: Add an iframe to your Angular component

In your Angular component where you want to display the Power BI report, add an **iframe** element to your template file. For example:

```
<iframe width="100%" height="500" frameborder="0" [src]="powerbiUrl"></iframe>
```

Here, the **powerbiUrl** variable will hold the embed URL for the Power BI report that you obtained in Step 2.

Step 4: Assign the embed URL to the iframe

In your Angular component TypeScript file, assign the embed URL to the `powerbiUrl` variable. You can do this in the `ngOnInit` method or in any other method where you want to display the report. For example:

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-report',
  templateUrl: './report.component.html'
})
export class ReportComponent implements OnInit {
  powerbiUrl: string;

  ngOnInit() {
    this.powerbiUrl = 'YOUR_REPORT_EMBED_URL';
  }
}
```

Here, replace `YOUR_REPORT_EMBED_URL` with the actual embed URL for your Power BI report that you obtained in Step 2.

Step 5: Test the Power BI report in the iframe

Run your Angular application and navigate to the component where you added the `iframe`. You should now see the Power BI report embedded in the `iframe`.

Note that the report will be displayed in the iframe with the same permissions as the user who is currently logged in to the Power BI service. If the user does not have permission to view the report, it will not be displayed in the iframe.

Also this approach has some limitations compared to using the Power BI JavaScript client library, such as limited customization options and less interactivity. Power BI client usage is described later in the document.

There are other ways which can provide more control on Power-bi reports and those requires some extra setup to be done before.

Now we are going to see how we can make use of Power BI Rest APIs

How to get power bi rest APIs

You can access the Power BI REST APIs by sending HTTP requests to the Power BI API endpoints.

Here are the steps you can follow to get started with the Power BI REST APIs:

1. Register an application in Azure Active Directory:

- To access the Power BI REST APIs, you need to register an application in Azure Active Directory (Azure AD). You can do this by following the instructions in the Microsoft documentation: [Register an app in Azure Active Directory](#)

2. Authenticate with Azure AD:

- To access the Power BI REST APIs, you need to obtain an access token from Azure AD. You can do this by following the instructions in the Microsoft documentation: [Authenticate with Azure AD](#)

3. Send HTTP requests to the Power BI REST API endpoints:

- Once you have obtained an access token, you can use it to send HTTP requests to the Power BI REST API endpoints. You can use any HTTP client library or tool to send the requests, such as `curl`, Postman, or the `HttpClient` module in Angular.
- The Power BI REST API endpoints allow you to perform a wide range of operations, such as getting a list of reports, getting report details, generating embed tokens, creating groups, and more. You can find the full list of endpoints and their descriptions in the Microsoft documentation: [Power BI REST API reference](#)

That's it! You should now be able to access the Power BI REST APIs and perform various operations on your Power BI resources. Note that some operations may require additional permissions or roles, so make sure to check the documentation for details.

Two important points to note here :

1. The Power BI REST APIs support both HTTP and HTTPS protocols.

However, it is recommended to use HTTPS to ensure that the communication between your application and the Power BI service is secure and encrypted. HTTPS also helps protect against unauthorized access and interception of data during transit.

When you make HTTP requests to the Power BI REST API endpoints, you should use the `https://api.powerbi.com` base URL instead of `http://api.powerbi.com`. For example:

```
https://api.powerbi.com/v1.0/myorg/reports
```

Here, `https://api.powerbi.com` is the base URL, and `/v1.0/myorg/reports` is the endpoint path.

In summary, while both HTTP and HTTPS are supported, it is recommended to use HTTPS to ensure secure communication between your application and the Power BI service.

2. The lifetime of an embed token for a Power BI report is determined by its expiration date and time.

By default, an embed token is valid for one hour. After one hour, the token expires and cannot be used to access the report. If you try to use an expired token to access a report, you will receive a 401 Unauthorized error.

You can set the expiration date and time of an embed token when you generate it using the Power BI REST API. You can set the expiration time to any value between 10 minutes and 1 year from the time the token was issued.

To ensure that your embedded reports remain secure, it is recommended to use short-lived embed tokens that expire after a short period of time. You should also use secure communication protocols such as HTTPS and implement other security measures such as authentication and authorization checks to ensure that only authorized users can access your embedded reports.

According to Microsoft's documentation, the Power BI JavaScript client library requires Angular version 13 or later.

Here is an updated version of the steps to show Power BI report in Angular application having version ≥ 13 :

1. Install the Power BI JavaScript client package:

- Install the Power BI JavaScript client package using npm. You can do this by running the following command in the terminal:

```
npm install powerbi-client
```

2. Import the Power BI JavaScript client:

- Import the Power BI JavaScript client in your component by adding the following import statement at the top of the file:

```
import * as pbi from 'powerbi-client';
```

3. Create a Power BI embed configuration:

- Create a Power BI embed configuration object with the embed token and report ID you obtained from Power BI service. You can also set additional options such as filters, page name, etc. For example:

```
const config: pbi.models.IEmbedConfiguration = {  
  type: 'report',  
  embedUrl: 'https://app.powerbi.com/reportEmbed?reportId=<reportId>',  
  tokenType: pbi.models.TokenType.Embed,  
  accessToken: '<embedToken>',  
  settings: {  
    filterPaneEnabled: false,  
    navContentPaneEnabled: true,  
    background: pbi.models.BackgroundType.Transparent,  
  },  
};
```

4. Embed the report:

- Embed the report in your component using the Power BI JavaScript client. For example:

```
const reportContainer = document.getElementById('reportContainer');  
const report = powerbi.embed(reportContainer, config);
```

Here, `reportContainer` is the HTML element where the report will be embedded.

5. Render the report:

- Finally, render the report by calling the `report.render()` method. For example:

```
report.render();
```

That's it! Your Power BI report should now be displayed in your Angular 13 application. You can customize the report's appearance and behavior by modifying the `config` object in step 3.