Connecting SPFx Projects to Data Sources

JS PADOAN

MICROSOFT CERTIFIED TRAINER

@JsPadoan https://www.linkedin.com/in/jspadoan

Overview

Connecting and using Microsoft Graph

- Graph APIs
- Graph Explorer
- Graph Toolkit

Using third party APIs

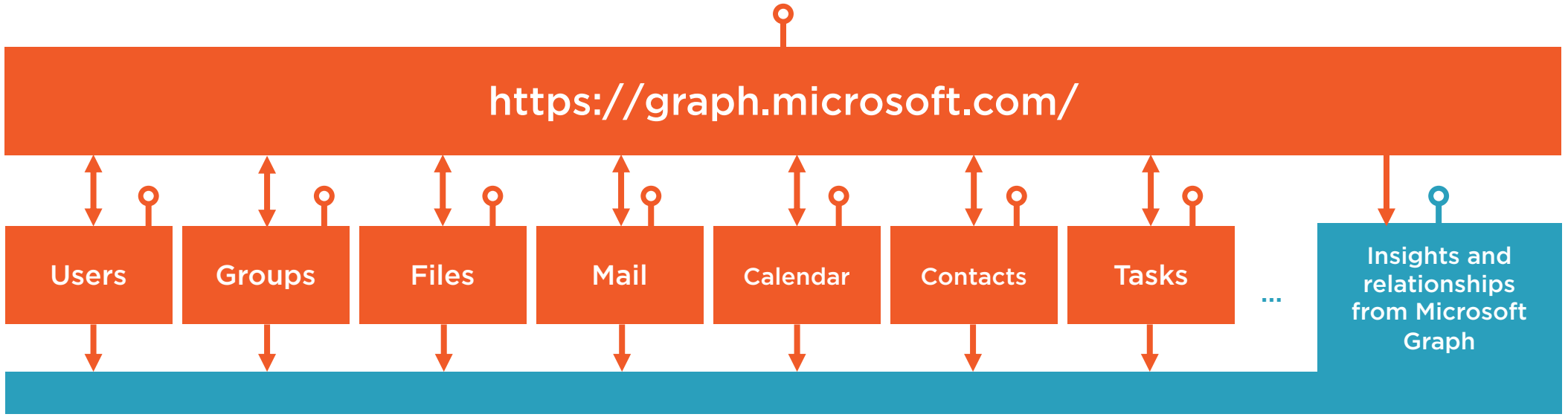
Establishing connections between SPFx Webparts

- Provider
- Consumer

Connecting and Using Microsoft Graph

Overview of Microsoft Graph

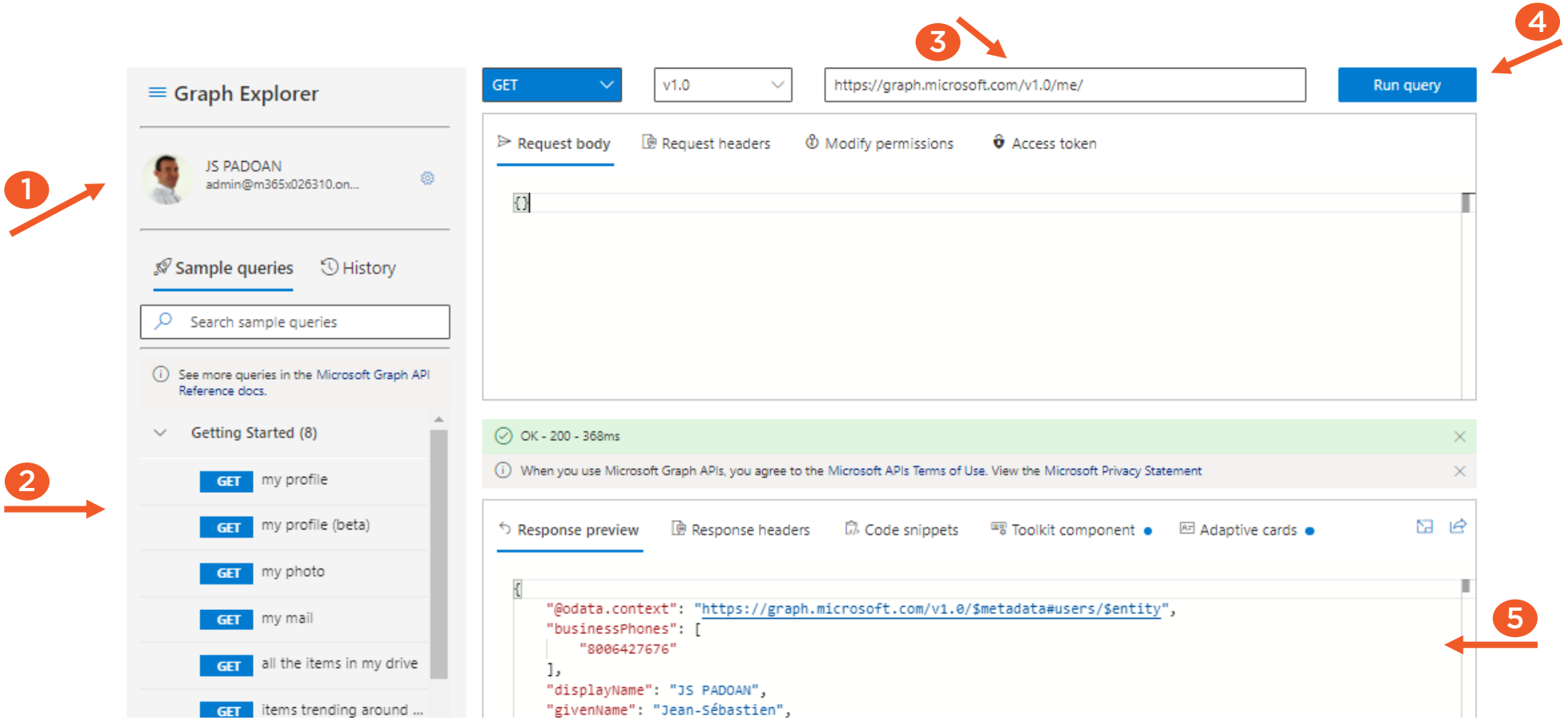
Overview of Graph APIs



https://graph.microsoft.com/v1.0/me/planner/tasks

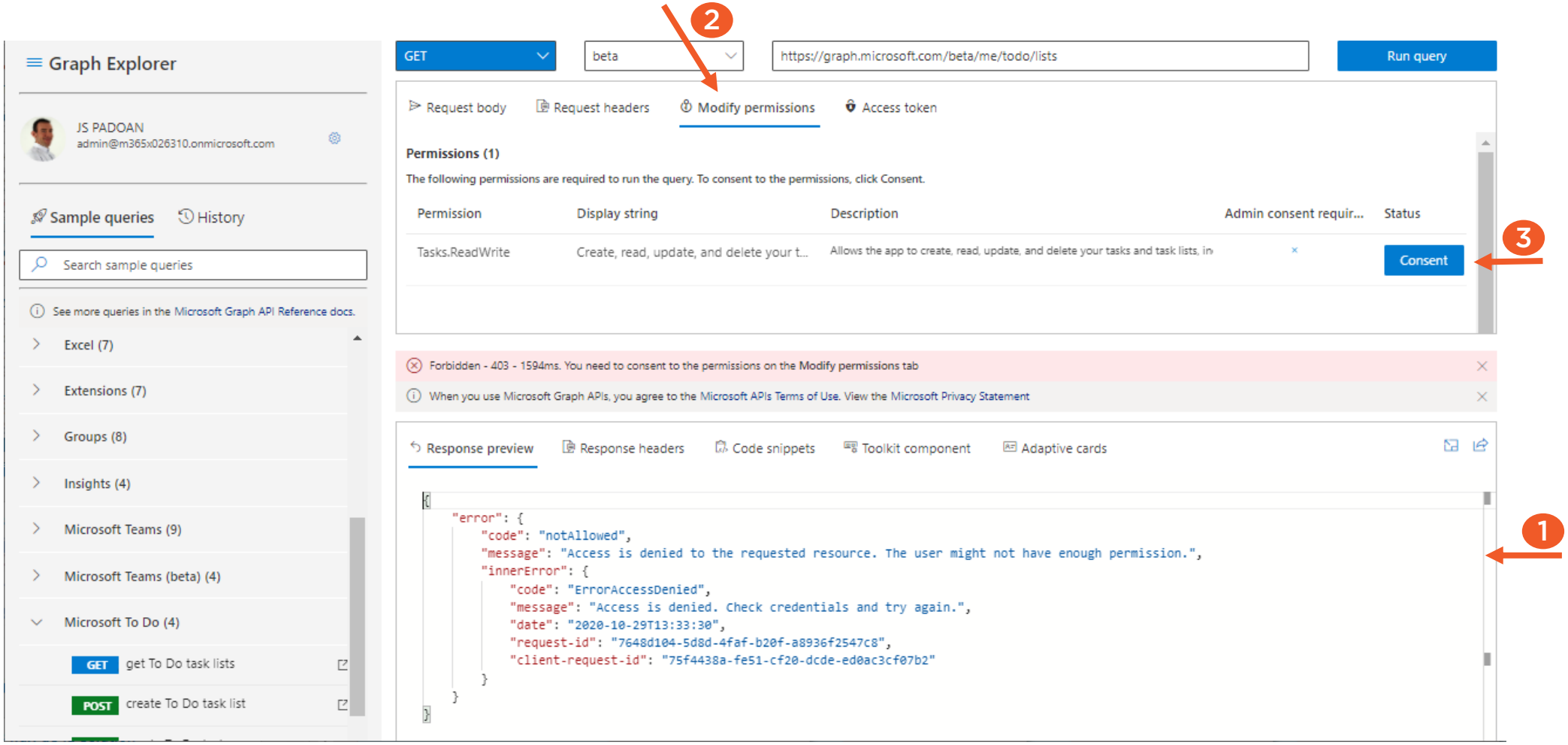
https://graph.microsoft.com/v1.0/me/drive/recent

Making Live Testing with Graph Explorer

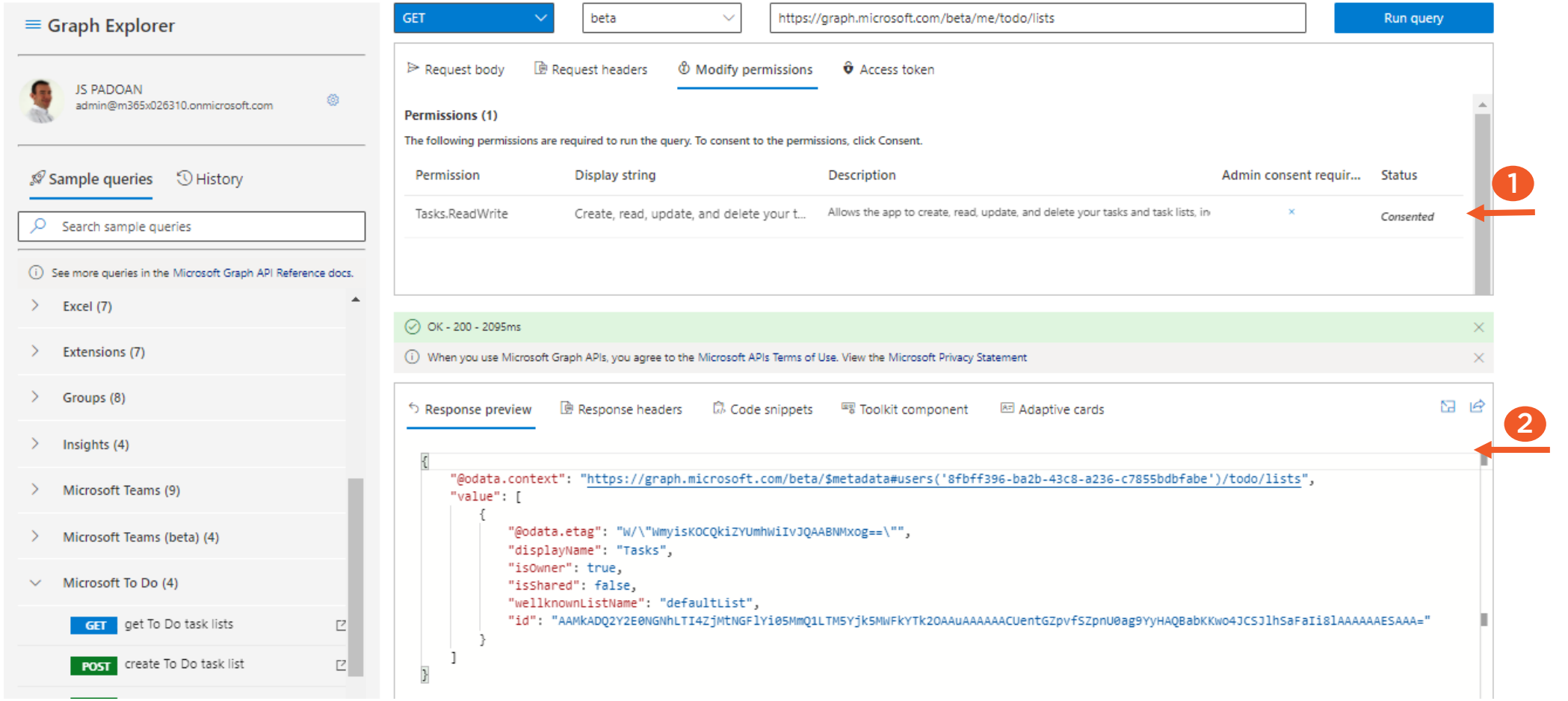


https://developer.microsoft.com/en-us/graph/graph-explorer

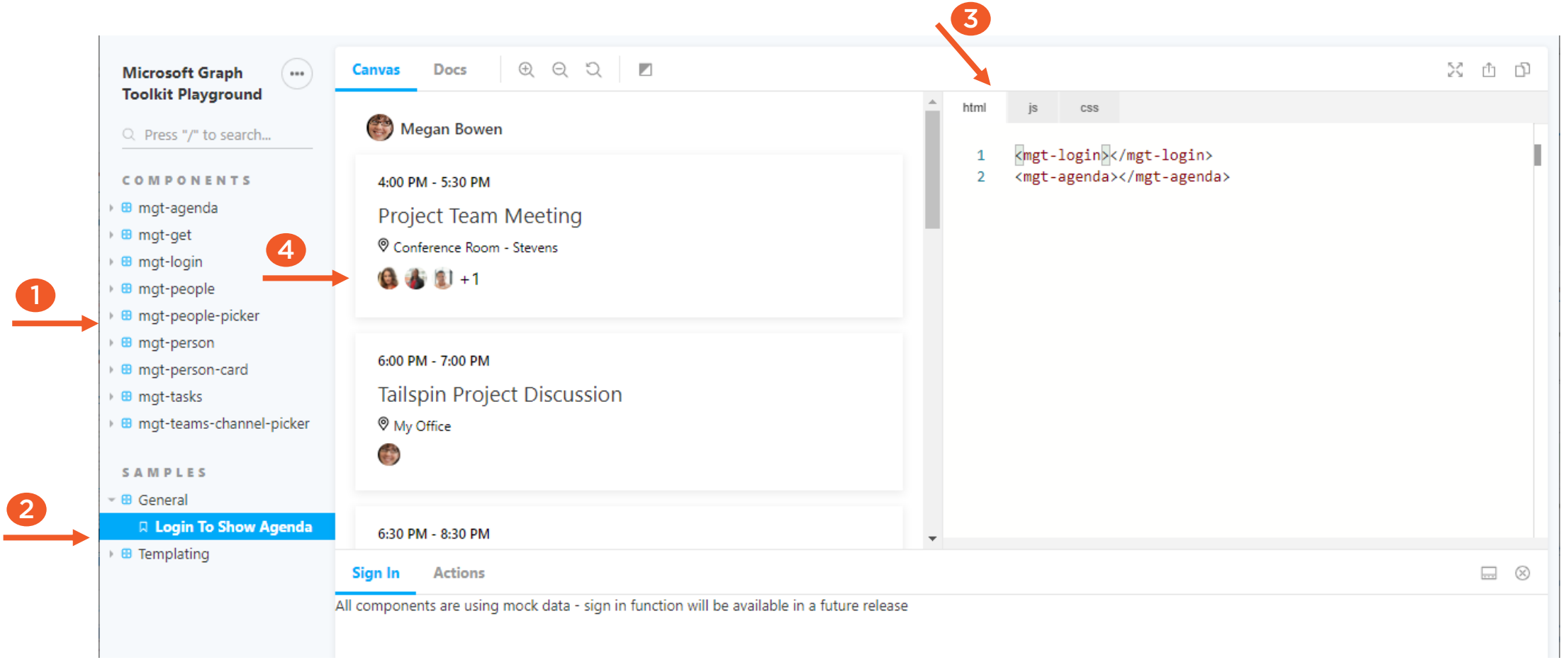
Making Live Testing with Graph Explorer



Making Live Testing with Graph Explorer



Using Microsoft Graph Toolkit



https://mgt.dev/

Using Graph APIs into SPFx Components

```
import { MSGraphClient } from '@microsoft/sp-http';

…

context.msGraphClientFactory
.getClient()
.then((c: MSGraphClient): void => {
    c.api('/me/joinedTeams').get((error, response: any, rawResponse?: any) => {
        this.setState({ nbTeams : Object.keys(response).length});
    });
});
```

1
2
3
4
5

Using Graph APIs into SPFx Components

```
import { MSGraphClient } from '@microsoft/sp-http';

import * as MicrosoftGraph from '@microsoft/microsoft-graph-types’;

…

context.msGraphClientFactory

.getClient()

.then((c: MSGraphClient): void => {

      c.api('/me').get((error, user: MicrosoftGraph.User, rawResponse?: any) => {

        this.setState({ profile : user});

      });

    });
```

1

2

3

Demo

Create a SPFx Webpart to display user details:

- Use Graph API
- Retrieve user profile details
- Retrieve the number of Teams the currently connected user belongs to

Using Third Party APIs

Using Third Party APIs in SPFx Components

```
import {
    HttpClient,
    HttpClientResponse
  } from "@microsoft/sp-http";

 …

private _prepareHeaders(): Headers {
        const requestHeaders: Headers = new Headers();
        requestHeaders.append("Accept", "application/json");
        requestHeaders.append("Content-Type", "application/json");
        requestHeaders.append("Cache-Control", "no-cache");
        requestHeaders.append("Ocp-Apim-Subscription-Key", "THEKEY");
        return requestHeaders;
      }

…
```

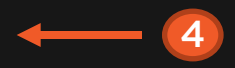
1

2

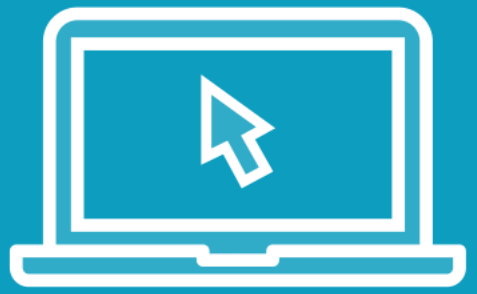
Using Third Party APIs in SPFx Components

```
const response: HttpClientResponse = await context.httpClient.post(
  ”URL”,
  HttpClient.configurations.v1,
  {
    body: JSON.stringify(body),
    headers: this._prepareHeaders()
  }
);

const responseJSON: any = await response.json();
```



Demo



Extract and display language and sentiment of comments

Modify previously created Webpart (ArticleComments)

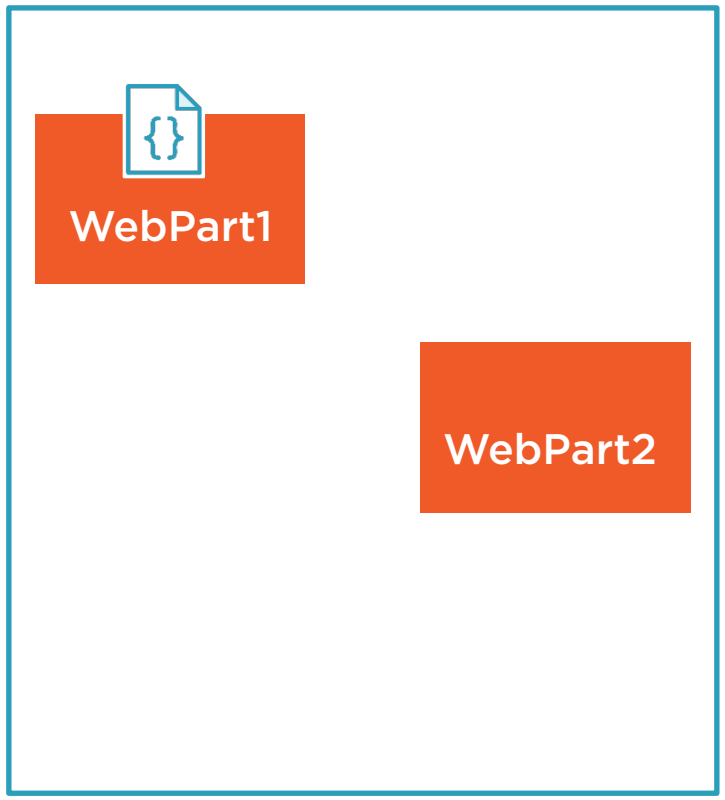
- Connect to SharePoint List API
- Retrieve comments on an article page

Call Microsoft Sentiment Analysis Cognitive API

- Test Cognitive API calls with Postman
- Modify the Webpart to include on-the-fly sentiment analysis

Establishing Connections between SPFx WebParts

Overview of Connecting Web Parts



Establish link between web parts on the same page

A link connects 1 origin to 1 destination

Multiple links can be made to the same origin

At a given time, data will be transmitted through established links

Overview of Connecting Web Parts

Creating Endpoints (Provider)

```
export interface IData { … }

…

import {
  IDynamicDataPropertyDefinition,
  IDynamicDataCallables, IDynamicDataAnnotatedPropertyValue
} from '@microsoft/sp-dynamic-data';
…
export default class myWebPart ... implements IDynamicDataCallables {

public getPropertyDefinitions(): ReadonlyArray<IDynamicDataPropertyDefinition> { }

public getPropertyValue(propertyId: string)  : IData { }

public getAnnotatedPropertyValue?(propertyId: string): IDynamicDataAnnotatedPropertyValue { }

…

// notify subscribers that 'TheProperty' has changed
WebPartContext.dynamicDataSourceManager.notifyPropertyChanged('TheProperty');

}
```

1 2 3 4 5 6 7

Consuming Endpoints

```
export interface IWebPartProps {
myProperty: DynamicProperty<IData>;
…
}

…

protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {
 …
 PropertyPaneDynamicFieldSet({
    label: 'Select event source’,
    fields: [ PropertyPaneDynamicField('myProperty’, {label: ’…’}) ]
 })
 …
}

public async componentDidUpdate?(…): Promise<void> {
    const data: IData = this.props.myProperty.tryGetValue();
…
}
```

1

2

3

Demo

Create Webparts and make them compliant to be connected:

- Create a Webpart (FilterNews) that could expose category of news
- Create a Webpart (SentimentNews) that could retrieve the category of news to search with Bing Search API
- Narrow the search with Sentiment Analysis

Connect the Webparts on a page to see the connection working

Summary

Connecting and using Microsoft Graph

- Graph APIs
- Graph Explorer
- Graph Toolkit

Using third party APIs

Establishing connections between SPFx web parts

- Provider
- Consumer