

Enhancing SPFx Projects



JS PADOAN

MICROSOFT CERTIFIED TRAINER

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



Overview



Enhancing SPFx components with JavaScript libraries

- Referencing JavaScript library
- Using external library
- Embed library in SPFx package
- Charting (with React component)

Increasing efficiency with SharePoint PnP

- Outline of Patterns and Practices
- Referencing and using PnP



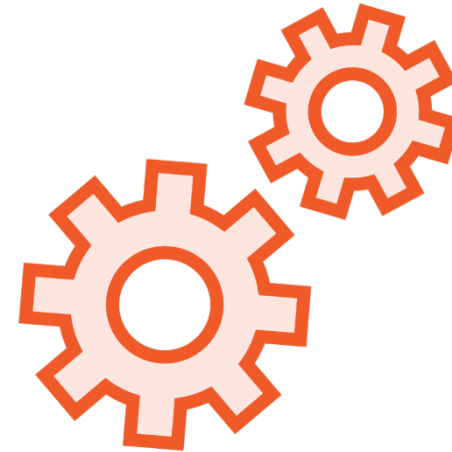
Enhancing SPFx Components with JavaScript Libraries



Referencing JavaScript Library



External component



**Component included
in SPFx package**



Using a JavaScript Library (jQuery) as External

```
"externals": {  
  "jquery": {  
    "path": "https://code.jquery.com/jquery-VERSION.min.js",  
    "globalName": "jQuery"  
  }  
}
```

```
npm install --save @types/jquery
```

```
import * as $ from 'jquery';
```

```
Use $ in your code !
```



Using a Package Included JavaScript Library

```
npm install YourComponent --save
```



```
npm install --save @types/jquery
```



```
require('node_modules/YourComponent/YourComponent.js');
```



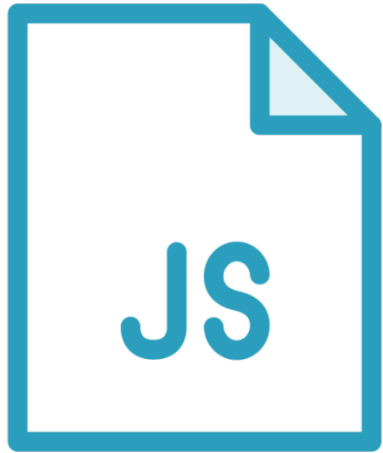
```
require('node_modules/YourComponent/YourComponent.css');
```



```
($('.class') as any).theMethod(theParams);
```



Creating Charts in SPFx Components



Use JavaScript library



Example with D3.js

```
npm install d3 --save
```



```
npm install @types/d3 --save
```



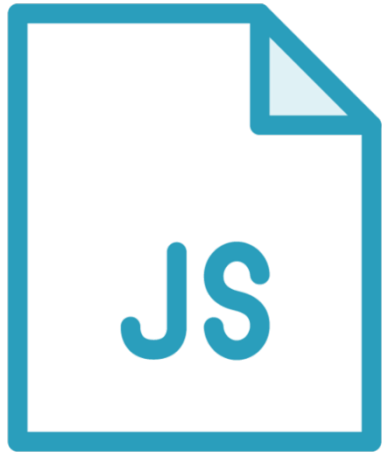
```
import * as d3 from 'd3';
```



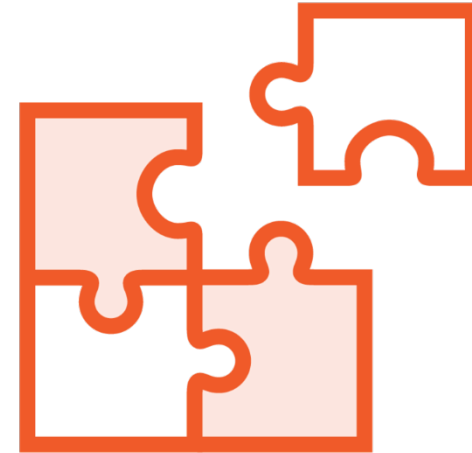
```
d3.select("#TheDiv").append("svg").....
```



Charts in SPFx Components



Use JavaScript library



Use React component

<https://pnp.github.io/sp-dev-fx-controls-react/controls/ChartControl/>



Example with ChartControl

```
npm install @pnp/spfx-controls-react --save
```



```
import { ChartControl, ChartType } from '@pnp/spfx-controls-react/lib/ChartControl';
```



```
<ChartControl  
  type={ChartType.Bar}  
  data={{  
    labels: ['January', 'February', 'March', 'April', 'May', 'June', 'July'],  
    datasets: [{  
      label: 'My First dataset',  
      data: [65, 59, 80, 81, 56, 55, 40]  
    }]  
  }} />
```



Demo



Enhance “NewsSentiment” web part (Ep 1):

- Display the news as an attractive carousel
- Use jQuery as an external component
- Embed JavaScript component inside SPFx package

Enhance “NewsSentiment” web part (Ep 2):

- Display
- Use React component
- Upgrade the web part to display an optional donut chart



Increasing Efficiency with SharePoint PnP



Overview of Microsoft 365 PnP



Guidance

Samples and solutions

SDKs

Tools

<https://pnp.github.io/>



Referencing and Using PnP Framework

```
npm install @pnp/pnpjs -save
```



```
import { sp } from "@pnp/sp/presets/all";
```



```
sp.setup({  
  spfxContext: this.props.context  
});
```



```
await sp.web.lists.getByTitle("Site Pages")  
  .items  
  .select("Title, FileRef, EncodedAbsUrl")  
  .orderBy("Modified", false)  
  .top(1)  
  .get();
```



Demo



Upgrade “ArticleComments” web part:

- Display (using Fluent UI) the overall positive sentiment of that comments
- Install and reference PnP
- Retrieve most recent page and its comments using PnP



Summary



Enhancing SPFx components with JavaScript libraries

- Referencing JavaScript library
- Using external library
- Embed library in SPFx package
- Charting (with React component)

Increasing Efficiency with SharePoint PnP

- Outline of Patterns and Practices
- Referencing and using PnP

