SYMPRAXIS CONSULTING

Build Solutions for Microsoft 365 with a Fluent API Library

Julie Turner

Julie Turner

Partner/CTO

Working with SharePoint since 2007

Microsoft MVP, Office Apps and Services since 2017

Microsoft 365 PnP Team since 2019

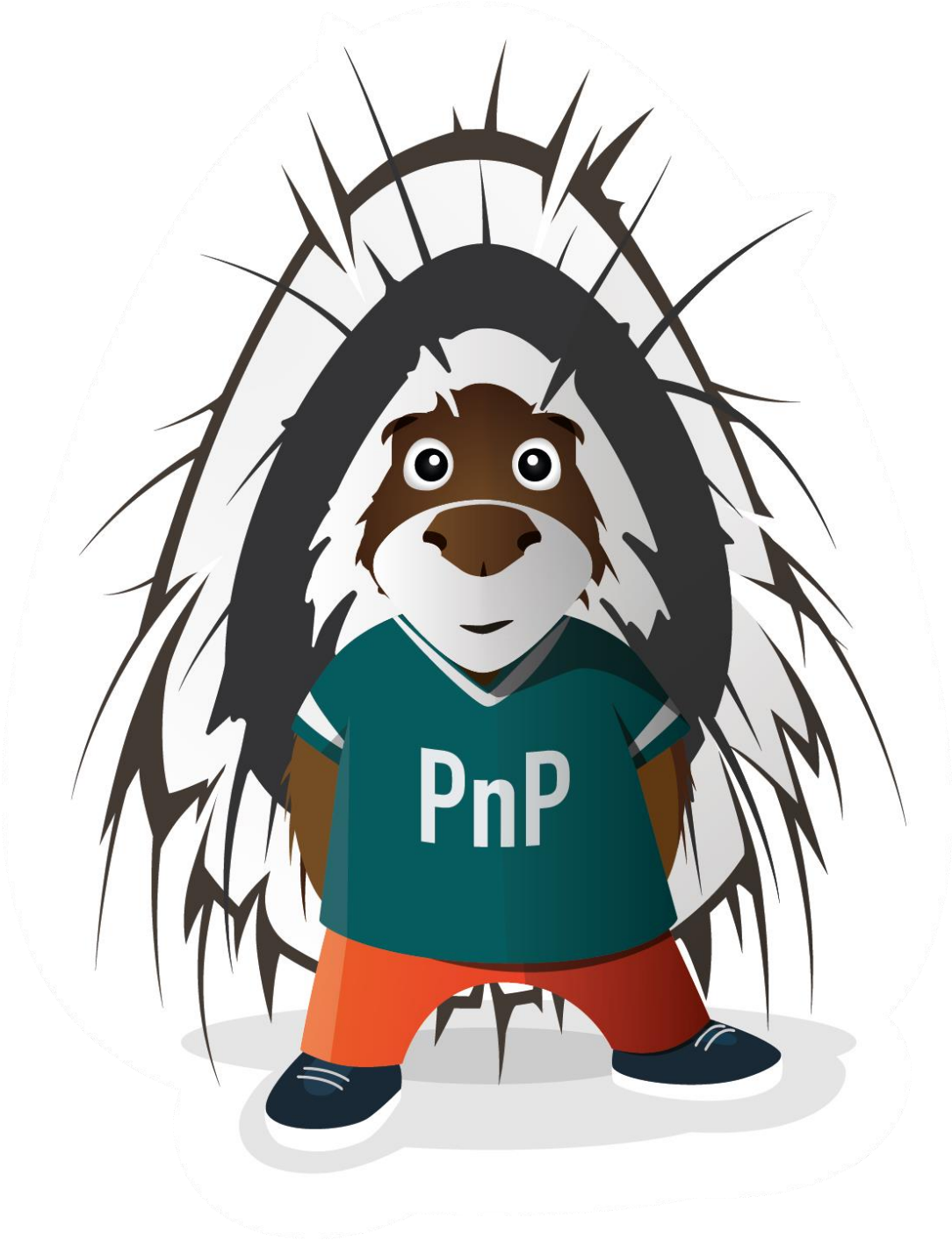
Open-source project co-maintainer: PnPjs & hTWOo

https://julieturner.net/me

SYMPRAXIS CONSULTING

Agenda

- PnP(js) Introduction
- Demos
  - Installation
  - Establish Context
  - Get Data
  - Caching
  - Batching
  - Cross Site Calls
- What's changed for V4
- Q&A



Microsoft 365 &
Power Platform Community

BLOG COMMUNITY CALLS GUIDANCE SAMPLES & SOLUTIONS SDKS TOOLS

Microsoft 365 & Power Platform Community

Learn from others how to build apps on Microsoft 365 & Power Platform.

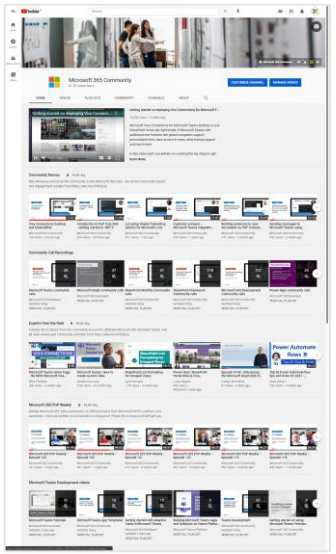
Don't reinvent the wheel. Focus on what truly matters for your organization.

Changing the world one contribution at a time!

SEE INITIATIVES →

https://aka.ms/community/home

Microsoft 365 & Power Platform community videos



aka.ms/community/videos

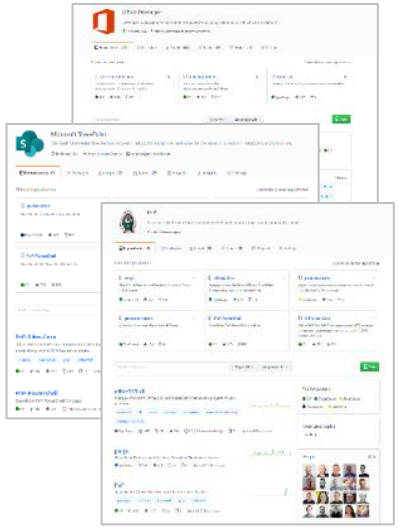
Subscribe today!

LinkedIn group for discussions

aka.ms/community/Li

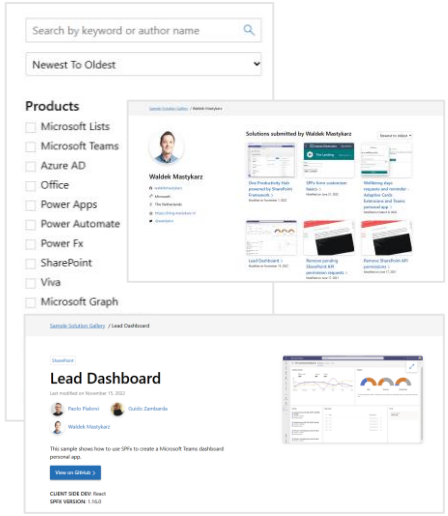
Join today!

Open-source



github.com/pnp
github.com/officedev
github.com/sharepoint
github.com/microsoftgraph

Sample galleries



aka.ms/community/samples
aka.ms/teams-samples
aka.ms/spfx-webparts
aka.ms/spfx-extensions
aka.ms/powerplatform-samples
aka.ms/list-formatting

Open-source initiatives and samples

aka.ms/community/home

What is PnPjs

Co-Maintainers

Beau Cameron
Practice Lead / Sr. Principal Architect
DMI

Patrick Rogers
Principal Product Manager
Microsoft

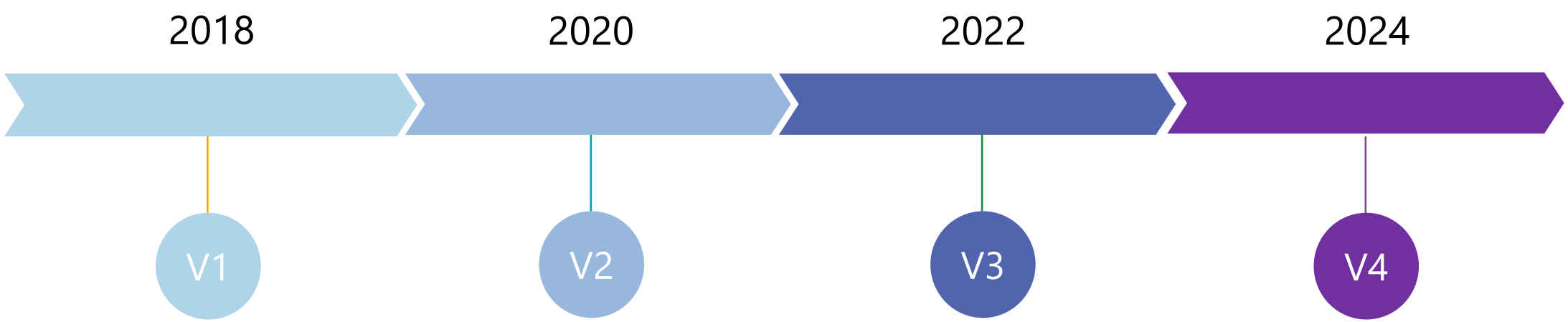
OPEN SOURCE INITIATIVE WHOLLY CREATED AND MAINTAINED BY THE COMMUNITY

FLUENT LIBRARIES FOR ACCESSING SHAREPOINT AND MICROSOFT GRAPH REST APIS IN A TYPE-SAFE WAY.

USED IN MORE THAN 37K+ TENANTS, GENERATING 37B+ REQUESTS EACH MONTH

ACCEPTS AND ENCOURAGES CONTRIBUTIONS AND CONSTRUCTIVE FEEDBACK FROM THE COMMUNITY.

A brief history



Focused on SharePoint APIs

Basic Graph Support

Introduced MSAL node/browser

Isolated runtimes

Complete re-write

Timelines, Queryables, Behaviors

Nightly builds

Improved composability

New Build System

Improved Microsoft Graph support

Iterators for Paging

Why use PnPjs?

Cuts development time

Fluent interface improves development and reduces errors

Reusable, but extensible

Developers can extend the native features with the extensibility model

One library, cross platform

Leverage the library across different scenarios without learning different frameworks

Traditional Approach

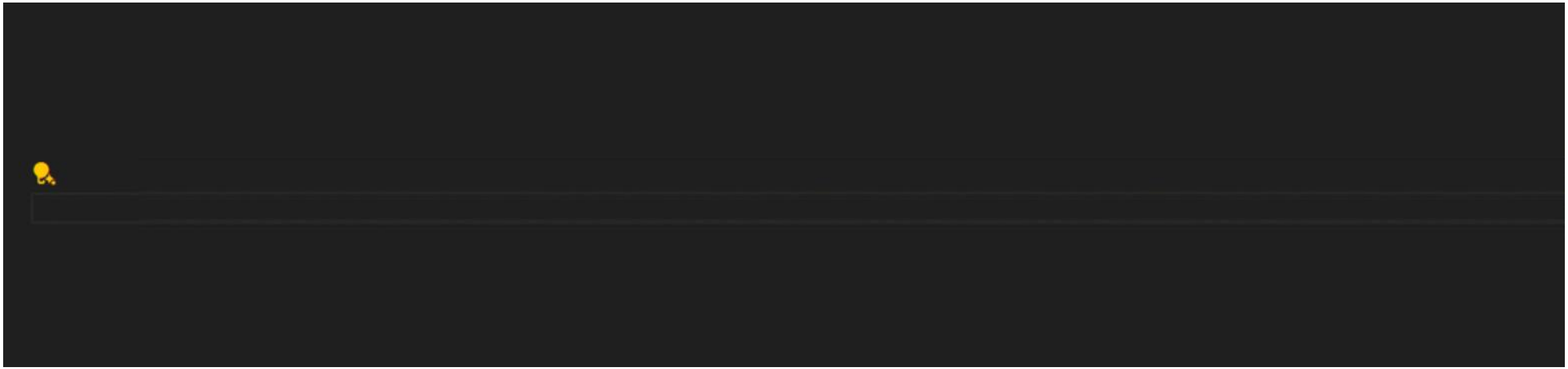
When writing API calls to SharePoint

string concatenation

```
api/web/lists/getbytitle('My List').items?$select=Title,Author/Id$expand=Author&$top=50
```

PnPjs - A fluent interface

Design method that relies on chaining



_api/web/lists/getbytitle('My List').items?$select=Title,Author/Id&$expand=Author&$top=50

What APIs do we support?

+

SharePoint

Attachments
Client-side pages
Column defaults
Comments and likes
Content types
Context info
Favorites
Features
Fields
Files
Folders
Forms
Group Site Manager
Hubsites
List Items
Lists
Navigation
Permissions
Profiles
Recycle Bin
Regional Settings
Related Items
Search
Security
Sharing
SharePoint Admin
Site Designs
Site Groups
Site Scripts
Site Users
Sites
Social
SP.Publishing.SitePageService
SP.Utilities.Utility
Subscriptions
Taxonomy
Tenant Properties
User Custom Actions
Views
Webs

Microsoft Graph

Admin
Analytics
App Catalog
Attachments
Bookings
Calendars
Cloud-Communications
Columns
Compliance
Contacts
Content Types
Conversations
Directory Objects
Files
Groups
Insights
Invitations
List items
Lists
Mail
Members
OneNote
Operations
Permissions
Photos
Places
Planner
Search
Shares
Subscriptions
Taxonomy
Teams
To-do
Users

Extending PnPjs

We expose methods for you to extend Graph & SharePoint APIs

SharePoint

- spGet
- spPost
- spDelete
- spPatch
- spPostMerge
- spPostDelete
- spPostDeleteEtag

Graph

- graphGet
- graphPost
- graphDelete
- graphPatch
- graphPut

```
const queryable = SPQueryable("https://cameroncode.sharepoint.com/sites/contosogroup8/_api/web/lists/getByTitle('SPFx List')/items(1)")
    .using(AssignFrom(sp.web));
const itemWithIdOf1 = await spGet(queryable);
console.log(itemWithIdOf1);
```

Authentication made *easier*

PnPjs includes built-in Auth Providers



SPFx

- SharePoint
- Microsoft Graph
- Auth Token
- MSAL

Browser

- MSAL

Node

- MSAL Nodejs
- Azure Identity

Getting Started

SharePoint Framework Code Demo

Packages

core	Provides shared functionality across all pnp libraries
queryable	Provides shared query functionality and base classes
sp	Provides a fluent api for working with SharePoint REST
graph	Provides a fluent api for working with Microsoft Graph
msaljsclient	Provides an msal wrapper suitable for use with PnPjs
nodejs	Provides functionality enabling the @pnp libraries within nodejs
azidjsclient	Provides an Azure Identity wrapper suitable for use with PnPjs
sp-admin	Provides a fluent api for working with M365 Tenant admin methods
logging	Light-weight, subscribable logging framework

npm install @pnp/sp @pnp/graph ...

package.json

{} package.json > ...

```
{
  "name": "pnpjs-demo",
  "version": "1.0.0",
  "private": true,
  "main": "lib/index.js",
  "engines": {
    "node": ">=16.13.0 <17.0.0 || >=18.17.1 <19.0.0"
  },
  Debug
  "scripts": {
    "build": "gulp bundle",
    "clean": "gulp clean",
    "test": "gulp test"
  },
  "dependencies": {
    "react": "17.0.1",
    "react-dom": "17.0.1",
    "@microsoft/sp-core-library": "1.18.2",
    "@microsoft/sp-property-pane": "1.18.2",
    "@microsoft/sp-webpart-base": "1.18.2",
    "@microsoft/sp-lodash-subset": "1.18.2",
    "@microsoft/sp-office-ui-fabric-core": "1.18.2",
    "@microsoft/sp-adaptive-card-extension-base": "1.18.2",
    "tslib": "2.3.1",
    "@pnp/sp": "^4.0.0",
    "@pnp/graph": "^4.0.0",
    "@pnp/logging": "^4.0.0",
    "@fluentui/react":"8.106.4"
  },
  "devDependencies": {
    "@types/react": "17.0.45",
```

p pnpjsteam

Home
Conversations
Teams
Documents
Notebook
Pages
Recent
pnplist
TestLib
Site contents
Recycle bin
Edit

PnPjs Demo

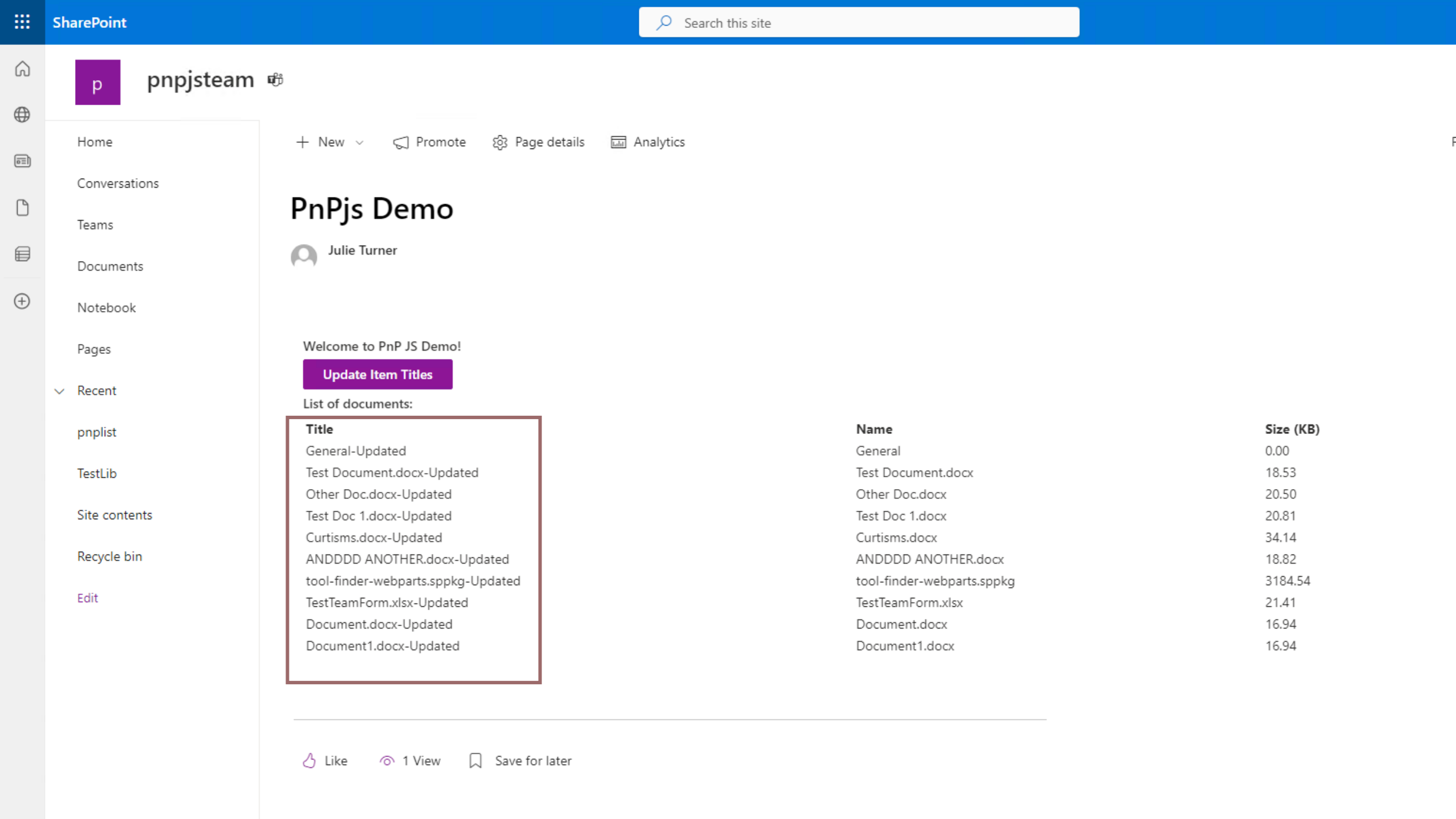
Julie Turner

Welcome to PnP JS Demo!

Update Item Titles

List of documents:

Title	Name	Size (KB)
General	General	0.00
Test Document	Test Document.docx	18.53
Other Doc	Other Doc.docx	20.50
Test Doc 1	Test Doc 1.docx	20.81
Curtisms	Curtisms.docx	34.14
ANDDDD ANOTHER	ANDDDD ANOTHER.docx	18.82
tool-finder-webparts	tool-finder-webparts.sppkg	3184.54
TestTeamForm	TestTeamForm.xlsx	21.41
Document	Document.docx	16.94
Document1	Document1.docx	16.94


SharePoint
Search this site
p
pnpjsteam
Home
Conversations
Teams
Documents
Notebook
Pages
Recent
pnplist
TestLib
Site contents
Recycle bin
Edit
New
Promote
Page details
Analytics
PnPjs Demo
Julie Turner
Welcome to PnP JS Demo!
Update Item Titles
List of documents:
Title
General-Updated
Test Document.docx-Updated
Other Doc.docx-Updated
Test Doc 1.docx-Updated
Curtisms.docx-Updated
ANDDDD ANOTHER.docx-Updated
tool-finder-webparts.sppkg-Updated
TestTeamForm.xlsx-Updated
Document.docx-Updated
Document1.docx-Updated
Name
General
Test Document.docx
Other Doc.docx
Test Doc 1.docx
Curtisms.docx
ANDDDD ANOTHER.docx
tool-finder-webparts.sppkg
TestTeamForm.xlsx
Document.docx
Document1.docx
Size (KB)
0.00
18.53
20.50
20.81
34.14
18.82
3184.54
21.41
16.94
16.94
Like
1 View
Save for later

Establishing Context

SharePoint Framework Code Demo

PnPjsWebPart.ts


```
1   import * as React from 'react';
2   import * as ReactDom from 'react-dom';
3   import { Version } from '@microsoft/sp-core-library';
4   import { BaseClientSideWebPart } from '@microsoft/sp-webpart-base';
5   import pnpjs, { IpnpjsProps } from './components/PnPjs';
6   import { getSP } from './pnpjsConfig';
7
8   export interface IpnpjsWebPartProps {
9     description: string;
10  }
11
12  export default class pnpjsWebPart extends BaseClientSideWebPart<IpnpjsWebPartProps> {
13
14    protected async onInit(): Promise<void> {
15      //Initialize our _sp object that we can then use in other packages without having to pass around the context.
16      //  Check out pnpjsConfig.ts for an example of a project setup file.
17      getSP(this.context);
18
19    }
20
21    public render(): void {
22      const element: React.ReactElement<IpnpjsProps> = React.createElement(
23        pnpjs, {}
24      );
```

pnpjsConfig.ts

```ts
/* eslint-disable no-var */
import { WebPartContext } from "@microsoft/sp-webpart-base";

// import pnp and pnp logging system
import { spfi, SPFI, SPFx as spSPFx } from "@pnp/sp";
import { graphfi, GraphFI, SPFx as graphSPFx } from "@pnp/graph";
import { LogLevel, PnPLogging } from "@pnp/logging";
import "@pnp/sp/webs";
import "@pnp/sp/lists";
import "@pnp/sp/items";
import "@pnp/sp/batching";

var _sp: SPFI = null;
var _graph: GraphFI = null;

export const getSP = (context?: WebPartContext): SPFI => {
  if (_sp === null) {
    //You must add the @pnp/logging package to include the PnPLogging behavior it is no longer a peer dependency
    // The LogLevel set's at what level a message will be written to the console
    _sp = spfi().using(spSPFx(context)).using(PnPLogging(LogLevel.Warning));
  }
  return _sp;
};

export const getGraph = (context?: WebPartContext): GraphFI => {
  if (_graph === null) {
    //You must add the @pnp/logging package to include the PnPLogging behavior it is no longer a peer dependency
    // The LogLevel set's at what level a message will be written to the console
```

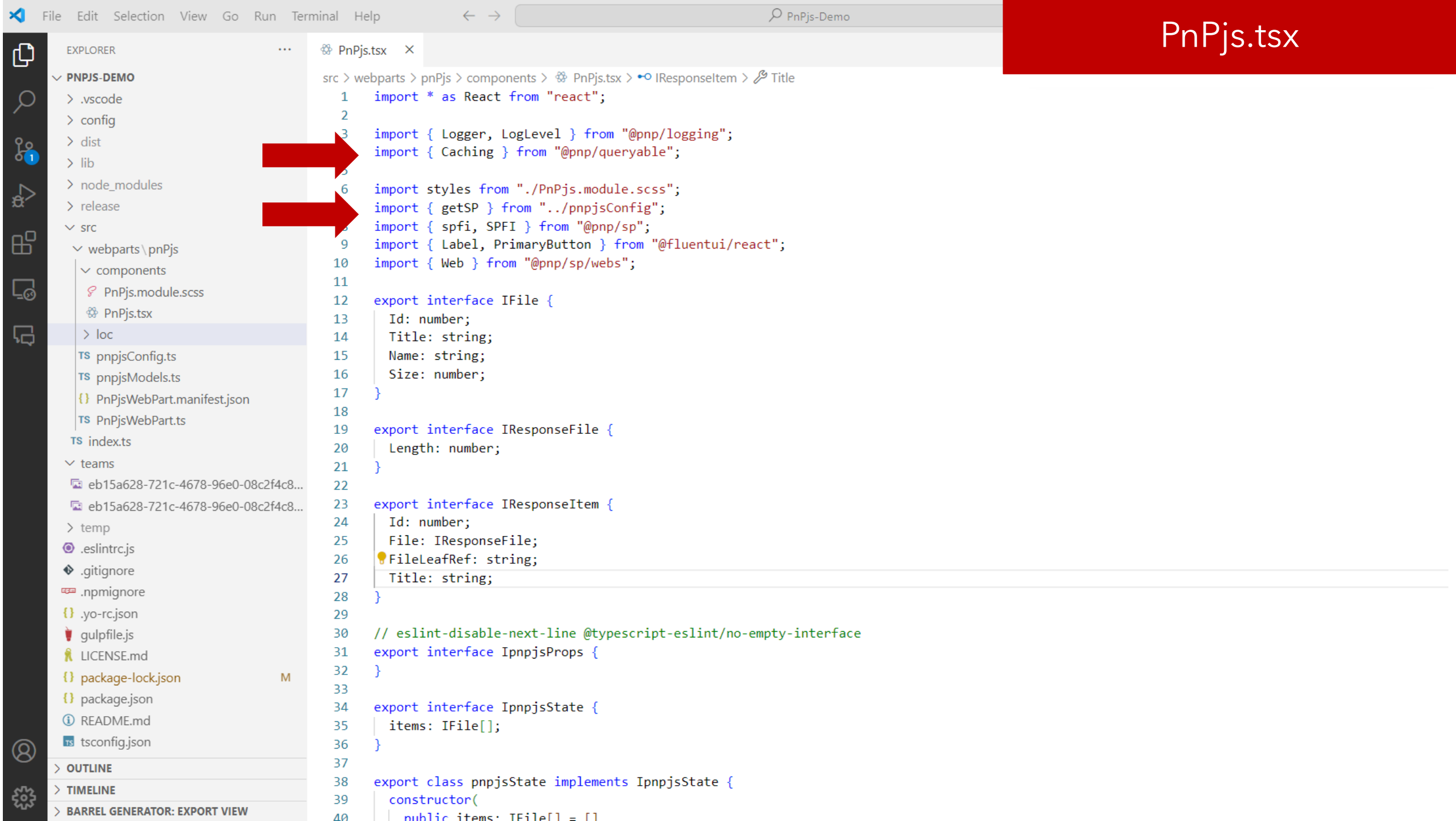
PnPjsWebPart.ts

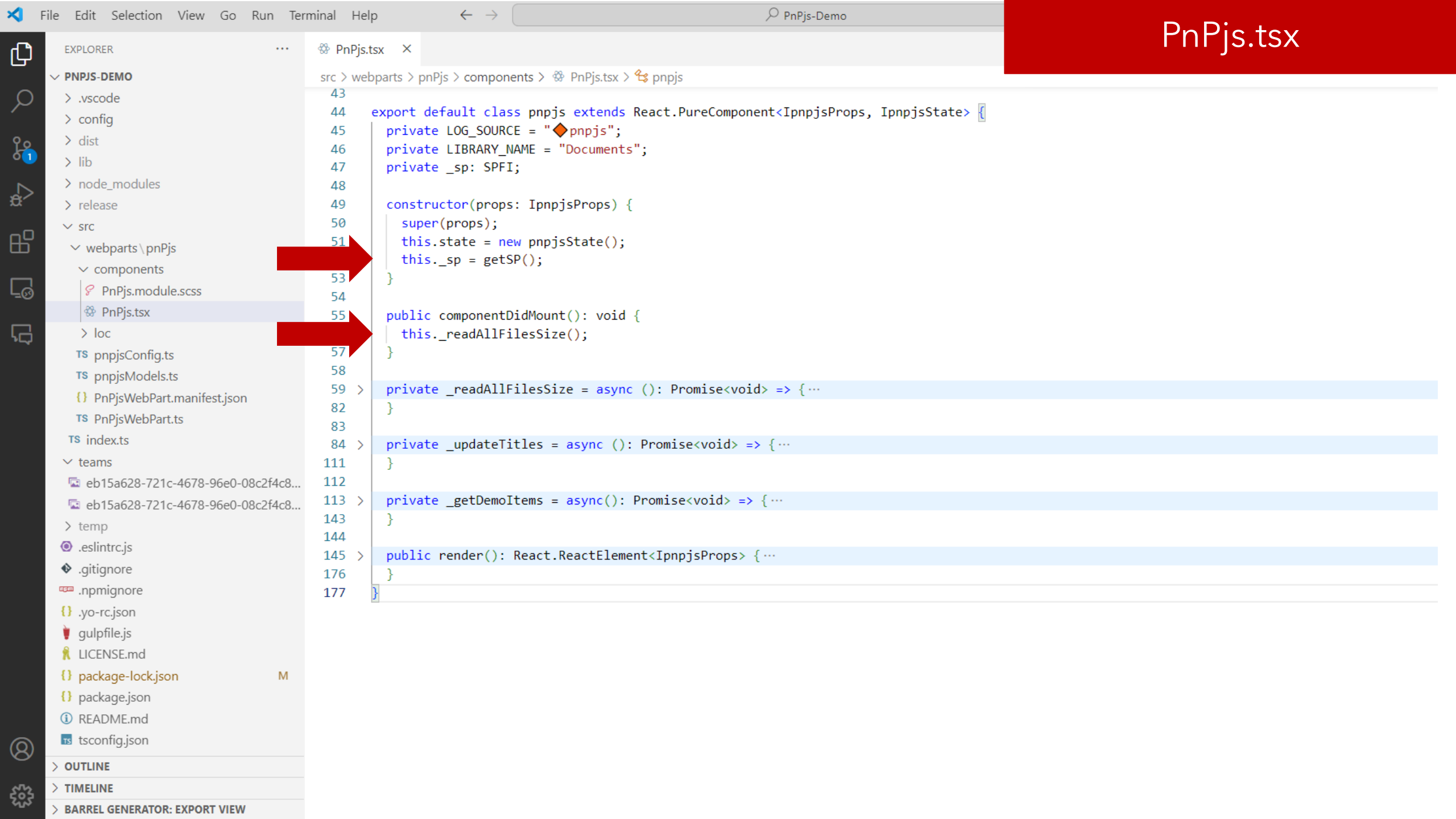
```
2   import * as ReactDom from 'react-dom';
3   import { Version } from '@microsoft/sp-core-library';
4   import { BaseClientSideWebPart } from '@microsoft/sp-webpart-base';
5   import pnpjs, { IpnpjsProps } from './components/PnPjs';
6   import { getSP } from './pnpjsConfig';
7
8   export interface IpnpjsWebPartProps {
9     description: string;
10  }
11
12  export default class pnpjsWebPart extends BaseClientSideWebPart<IpnpjsWebPartProps> {
13
14    protected async onInit(): Promise<void> {
15      //Initialize our _sp object that we can then use in other packages without having to pass around the context.
16      //  Check out pnpjsConfig.ts for an example of a project setup file.
17      getSP(this.context);
18
19    }
20
21    public render(): void {
22      const element: React.ReactElement<IpnpjsProps> = React.createElement(
23        pnpjs, {}
24      );
25
26      ReactDom.render(element, this.domElement);
27    }
28
29    protected onDispose(): void {
30      ReactDom.unmountComponentAtNode(this.domElement);
```

Getting Data, Caching, Batching

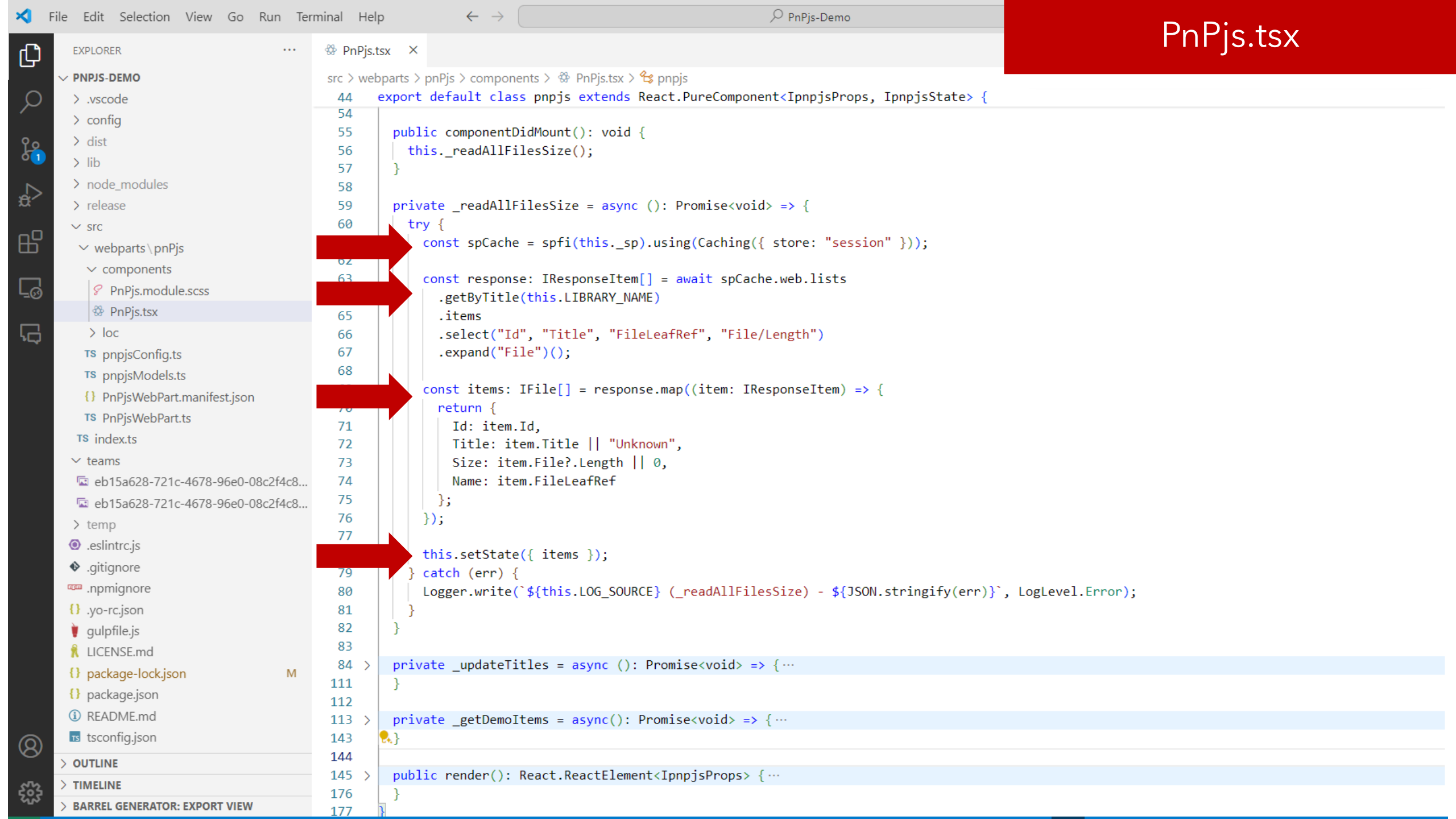
SharePoint Framework Code Demo

PnPjs.tsx

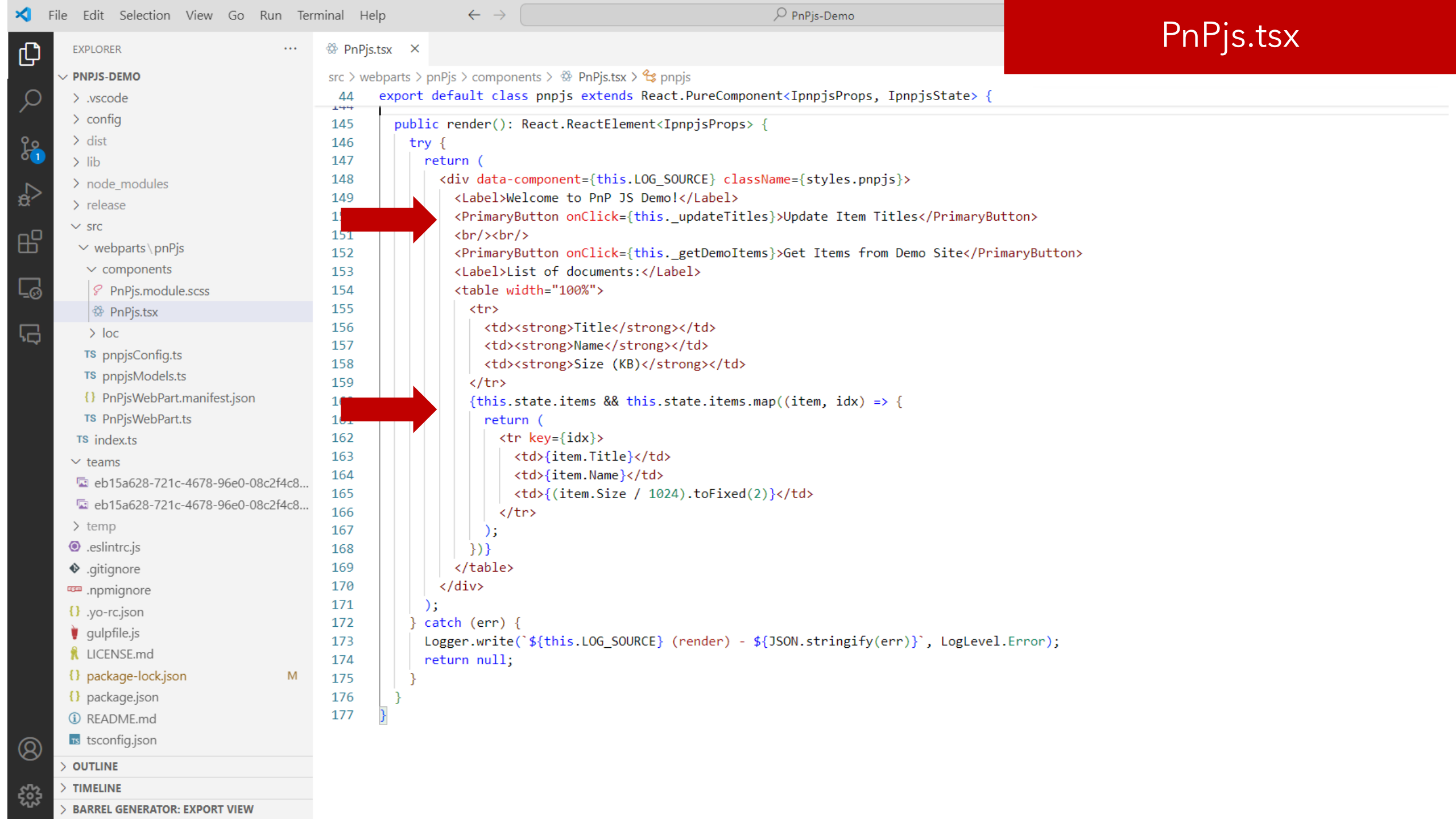

File Edit Selection View Go Run Terminal Help
PnPjs-Demo
EXPLORER
PNPJS-DEMO
.vscode
config
dist
lib
node_modules
release
src
webparts\pnPjs
components
PnPjs.module.scss
PnPjs.tsx
loc
pnpjsConfig.ts
pnpjsModels.ts
PnPjsWebPart.manifest.json
PnPjsWebPart.ts
index.ts
teams
eb15a628-721c-4678-96e0-08c2f4c8...
eb15a628-721c-4678-96e0-08c2f4c8...
temp
.eslintrc.js
.gitignore
.npmignore
.yo-rc.json
gulpfile.js
LICENSE.md
package-lock.json M
package.json
README.md
tsconfig.json
OUTLINE
TIMELINE
BARREL GENERATOR: EXPORT VIEW
PnPjs.tsx
src > webparts > pnPjs > components > PnPjs.tsx > IResponseItem > Title
1 import * as React from "react";
2
3 import { Logger, LogLevel } from "@pnp/logging";
4 import { Caching } from "@pnp/queryable";
5
6 import styles from "./PnPjs.module.scss";
7 import { getSP } from "../pnpjsConfig";
8 import { spfi, SPFI } from "@pnp/sp";
9 import { Label, PrimaryButton } from "@fluentui/react";
10 import { Web } from "@pnp/sp/webs";
11
12 export interface IFile {
13 Id: number;
14 Title: string;
15 Name: string;
16 Size: number;
17 }
18
19 export interface IResponseFile {
20 Length: number;
21 }
22
23 export interface IResponseItem {
24 Id: number;
25 File: IResponseFile;
26 FileLeafRef: string;
27 Title: string;
28 }
29
30 // eslint-disable-next-line @typescript-eslint/no-empty-interface
31 export interface IpnpjsProps {
32 }
33
34 export interface IpnpjsState {
35 items: IFile[];
36 }
37
38 export class pnpjsState implements IpnpjsState {
39 constructor(
40 public items: IFile[] = []


File Edit Selection View Go Run Terminal Help
PnPjs-Demo
PnPjs.tsx
EXPLORER
PNPJS-DEMO
.vscode
config
dist
lib
node_modules
release
src
webparts\pnPjs
components
PnPjs.module.scss
PnPjs.tsx
loc
pnpjsConfig.ts
pnpjsModels.ts
PnPjsWebPart.manifest.json
PnPjsWebPart.ts
index.ts
teams
eb15a628-721c-4678-96e0-08c2f4c8...
eb15a628-721c-4678-96e0-08c2f4c8...
temp
.eslintrc.js
.gitignore
.npmignore
.yo-rc.json
gulpfile.js
LICENSE.md
package-lock.json M
package.json
README.md
tsconfig.json
OUTLINE
TIMELINE
BARREL GENERATOR: EXPORT VIEW
src > webparts > pnPjs > components > PnPjs.tsx > pnpjs
43
44 export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsState> {
45 private LOG_SOURCE = "pnpjs";
46 private LIBRARY_NAME = "Documents";
47 private _sp: SPFI;
48
49 constructor(props: IpnpjsProps) {
50 super(props);
51 this.state = new pnpjsState();
this._sp = getSP();
53 }
54
55 public componentDidMount(): void {
this._readAllFilesSize();
57 }
58
59 private _readAllFilesSize = async (): Promise<void> => {
82 }
83
84 private _updateTitles = async (): Promise<void> => {
111 }
112
113 private _getDemoItems = async(): Promise<void> => {
143 }
144
145 public render(): React.ReactElement<IpnpjsProps> {
176 }
177 }

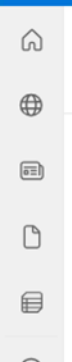
PnPjs.tsx


File Edit Selection View Go Run Terminal Help
PnPjs-Demo
EXPLORER
PNPJS-DEMO
.vscode
config
dist
lib
node_modules
release
src
webparts\pnPjs
components
PnPjs.module.scss
PnPjs.tsx
loc
pnpjsConfig.ts
pnpjsModels.ts
PnPjsWebPart.manifest.json
PnPjsWebPart.ts
index.ts
teams
eb15a628-721c-4678-96e0-08c2f4c8...
eb15a628-721c-4678-96e0-08c2f4c8...
temp
.eslintrc.js
.gitignore
.npmignore
.yo-rc.json
gulpfile.js
LICENSE.md
package-lock.json M
package.json
README.md
tsconfig.json
OUTLINE
TIMELINE
BARREL GENERATOR: EXPORT VIEW
PnPjs.tsx
src > webparts > pnPjs > components > PnPjs.tsx > pnpjs
export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsState> {
public componentDidMount(): void {
this._readAllFilesSize();
}
private _readAllFilesSize = async (): Promise<void> => {
try {
const spCache = spfi(this._sp).using(Caching({ store: "session" }));
const response: IResponseItem[] = await spCache.web.lists
.getByTitle(this.LIBRARY_NAME)
.items
.select("Id", "Title", "FileLeafRef", "File/Length")
.expand("File")();
const items: IFile[] = response.map((item: IResponseItem) => {
return {
Id: item.Id,
Title: item.Title || "Unknown",
Size: item.File?.Length || 0,
Name: item.FileLeafRef
};
});
this.setState({ items });
} catch (err) {
Logger.write(`${this.LOG_SOURCE} (_readAllFilesSize) - ${JSON.stringify(err)}`, LogLevel.Error);
}
}
private _updateTitles = async (): Promise<void> => {...
}
private _getDemoItems = async(): Promise<void> => {...
}
public render(): React.ReactElement<IpnpjsProps> {...
}
}

PnPjs.tsx


File Edit Selection View Go Run Terminal Help
PnPjs-Demo
EXPLORER
PNPJS-DEMO
.vscode
config
dist
lib
node_modules
release
src
webparts\pnPjs
components
PnPjs.module.scss
PnPjs.tsx
loc
pnpjsConfig.ts
pnpjsModels.ts
PnPjsWebPart.manifest.json
PnPjsWebPart.ts
index.ts
teams
eb15a628-721c-4678-96e0-08c2f4c8...
eb15a628-721c-4678-96e0-08c2f4c8...
temp
.eslintrc.js
.gitignore
.npmignore
.yo-rc.json
gulpfile.js
LICENSE.md
package-lock.json M
package.json
README.md
tsconfig.json
OUTLINE
TIMELINE
BARREL GENERATOR: EXPORT VIEW
PnPjs.tsx
src > webparts > pnPjs > components > PnPjs.tsx > pnpjs
44 export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsState> {
145 public render(): React.ReactElement<IpnpjsProps> {
146 try {
147 return (
148
149 <Label>Welcome to PnP JS Demo!</Label>
<PrimaryButton onClick={this._updateTitles}>Update Item Titles</PrimaryButton>
151

152 <PrimaryButton onClick={this._getDemoItems}>Get Items from Demo Site</PrimaryButton>
153 <Label>List of documents:</Label>
154 <table width="100%">
155 <tr>
156 <td>Title</td>
157 <td>Name</td>
158 <td>Size (KB)</td>
159 </tr>
{this.state.items && this.state.items.map((item, idx) => {
return (
162 <tr key={idx}>
163 <td>{item.Title}</td>
164 <td>{item.Name}</td>
165 <td>{(item.Size / 1024).toFixed(2)}</td>
166 </tr>
167);
168 })}
169 </table>
170
171);
172 } catch (err) {
173 Logger.write(`${this.LOG_SOURCE} (render) - ${JSON.stringify(err)}`, LogLevel.Error);
174 return null;
175 }
176 }
177 }



p pnpjsteam

Public group | Not following | 2 members

Home
Conversations
Teams
Documents
Notebook
Pages
Recent
pnplist
TestLib
Site contents
Recycle bin
Edit

PnPjs Demo

Julie Turner

Welcome to PnP JS Demo!

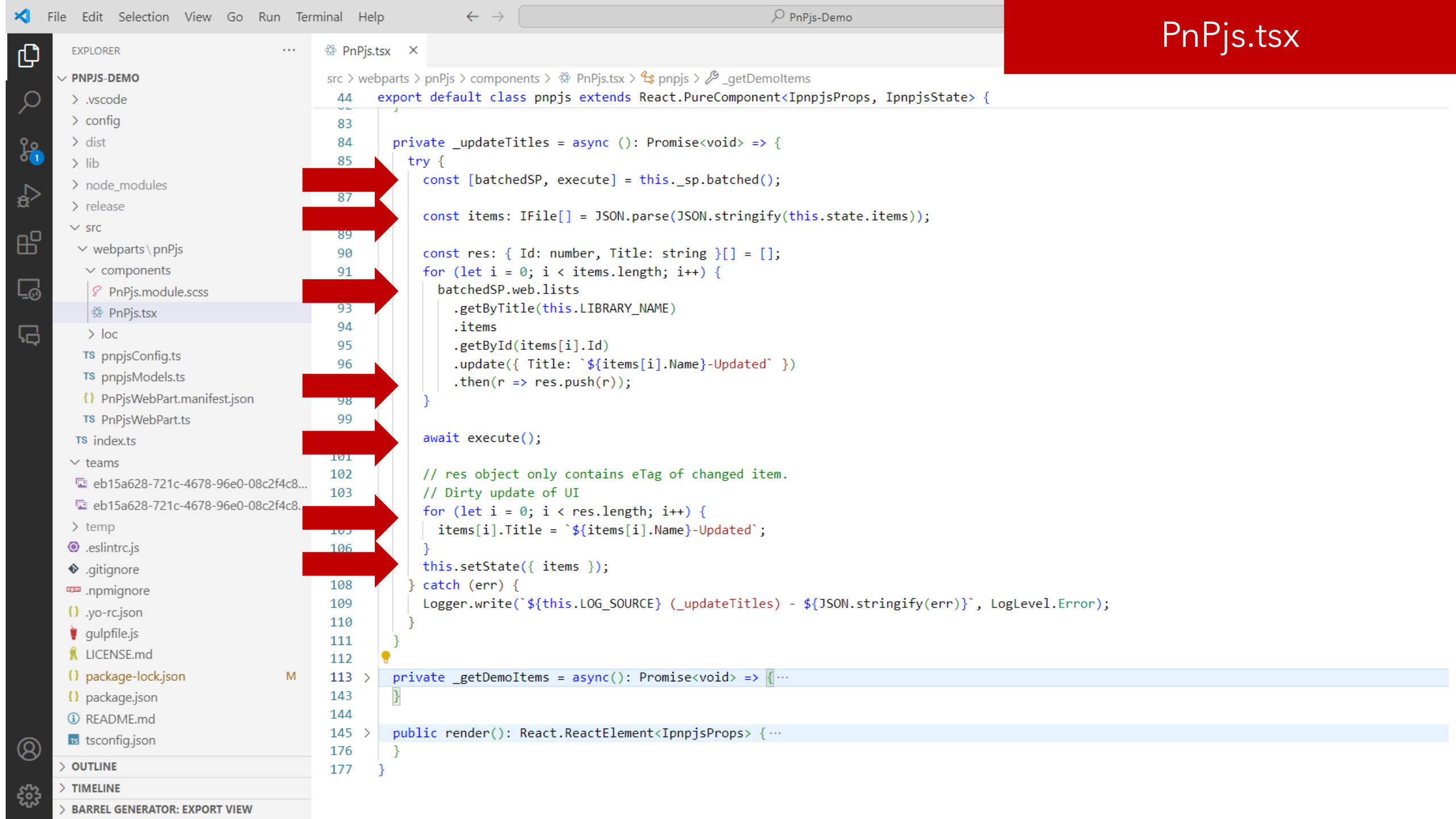
Update Item Titles

List of documents:

Title	Name	Size (KB)
General	General	0.00
Test Document	Test Document.docx	18.53
Other Doc	Other Doc.docx	20.50
Test Doc 1	Test Doc 1.docx	20.81
Curtisms	Curtisms.docx	34.14
ANDDDD ANOTHER	ANDDDD ANOTHER.docx	18.82
tool-finder-webparts	tool-finder-webparts.sppkg	3184.54
TestTeamForm	TestTeamForm.xlsx	21.41
Document	Document.docx	16.94
Document1	Document1.docx	16.94

Like | Save for later

PnPjs.tsx


File Edit Selection View Go Run Terminal Help
PnPjs-Demo
EXPLORER
PNPJS-DEMO
.vscode
config
dist
lib
node_modules
release
src
webparts\pnPjs
components
PnPjs.module.scss
PnPjs.tsx
loc
pnpjsConfig.ts
pnpjsModels.ts
PnPjsWebPart.manifest.json
PnPjsWebPart.ts
index.ts
teams
eb15a628-721c-4678-96e0-08c2f4c8...
eb15a628-721c-4678-96e0-08c2f4c8...
temp
.eslintrc.js
.gitignore
.npmignore
.yo-rc.json
gulpfile.js
LICENSE.md
package-lock.json M
package.json
README.md
tsconfig.json
OUTLINE
TIMELINE
BARREL GENERATOR: EXPORT VIEW
PnPjs.tsx
src > webparts > pnPjs > components > PnPjs.tsx > pnpjs > _getDemoItems
export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsState> {
private _updateTitles = async (): Promise<void> => {
try {
const [batchedSP, execute] = this._sp.batched();
const items: IFile[] = JSON.parse(JSON.stringify(this.state.items));
const res: { Id: number, Title: string }[] = [];
for (let i = 0; i < items.length; i++) {
batchedSP.web.lists
.getByTitle(this.LIBRARY_NAME)
.items
.getById(items[i].Id)
.update({ Title: `${items[i].Name}-Updated` })
.then(r => res.push(r));
}
await execute();
// res object only contains eTag of changed item.
// Dirty update of UI
for (let i = 0; i < res.length; i++) {
items[i].Title = `${items[i].Name}-Updated`;
}
this.setState({ items });
} catch (err) {
Logger.write(`${this.LOG_SOURCE} (_updateTitles) - ${JSON.stringify(err)}`, LogLevel.Error);
}
}
private _getDemoItems = async(): Promise<void> => {...
}
public render(): React.ReactElement<IpnpjsProps> {...
}
}

Cross Site Calls

SharePoint Framework Code Demo

p pnpjsteam

Home
Conversations
Teams
Documents
Notebook
Pages
Recent
pnplist
TestLib
Site contents
Recycle bin
Edit

PnPjs Demo

Julie Turner

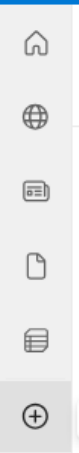
Welcome to PnP JS Demo!

Update Item Titles

Get Items from Demo Site

List of documents:

Title	Name	Size (KB)
General-Updated	General	0.00
Test Document.docx-Updated	Test Document.docx	18.53
Other Doc.docx-Updated	Other Doc.docx	20.50
Test Doc 1.docx-Updated	Test Doc 1.docx	20.81
Curtisms.docx-Updated	Curtisms.docx	34.14
ANDDDD ANOTHER.docx-Updated	ANDDDD ANOTHER.docx	18.82
tool-finder-webparts.sppkg-Updated	tool-finder-webparts.sppkg	3184.54
TestTeamForm.xlsx-Updated	TestTeamForm.xlsx	21.41
Document.docx-Updated	Document.docx	16.94
Document1.docx-Updated	Document1.docx	16.94



p pnpjsteam

Public group

Home
Conversations
Teams
Documents
Create
ebook
Pages
Recent
pnplist
TestLib
Site contents
Recycle bin
Edit

PnPjs Demo

Julie Turner

Welcome to PnP JS Demo!

Update Item Titles

Get Items from Demo Site

List of documents:

Title	Name	Size (KB)
General-Updated	General	0.00
Test Document.docx-Updated	Test Document.docx	18.53
Other Doc.docx-Updated	Other Doc.docx	20.50
Test Doc 1.docx-Updated	Test Doc 1.docx	20.81
Curtisms.docx-Updated	Curtisms.docx	34.14
ANDDDD ANOTHER.docx-Updated	ANDDDD ANOTHER.docx	18.82
tool-finder-webparts.sppkg-Updated	tool-finder-webparts.sppkg	3184.54
TestTeamForm.xlsx-Updated	TestTeamForm.xlsx	21.41
Document.docx-Updated	Document.docx	16.94
Document1.docx-Updated	Document1.docx	16.94
Unknown	123456	0.00
WidgetPricingSheet.xlsx-Updated	WidgetPricingSheet.xlsx	17.08
AutomatedModel.xlsx-Updated	AutomatedModel.xlsx	23.59
Unknown	SiteTestTemplate.dotx	19.15
Unknown	PnPTeam Site.url	0.09
Unknown	Test.docx	18.90
Unknown	Test.docx	17.61

src > webparts > pnPjs > components > PnPjs.tsx > pnpjs > _updateTitles

PnPjs.tsx

```
44   export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsS
84     private _updateTitles = async (): Promise<void> => {
110    }
111  }
112
113  private _getDemoItems = async(): Promise<void> => {
       try {…
137    } catch (err) {
138      Logger.write(`${this.LOG_SOURCE} (_getDemoItems) - ${JSON.stringify(err)}`, LogLevel.Error);
139    }
140  }
141
142  public render(): React.ReactElement<IpnpjsProps> {
143    try {
144      return (
145        <div data-component={this.LOG_SOURCE} className={styles.pnpjs}>
146          <Label>Welcome to PnP JS Demo!</Label>
147          <PrimaryButton onClick={this._updateTitles}>Update Item Titles</PrimaryButton>
148          <br/><br/>
149          <PrimaryButton onClick={this._getDemoItems}>Get Items from Demo Site</PrimaryButton>
150          <Label>List of documents:</Label>
151          <table width="100%">
152            <tr>
153              <td><strong>Title</strong></td>
154              <td><strong>Name</strong></td>
155              <td><strong>Size (KB)</strong></td>
156            </tr>
```

PnPjs.tsx

```
44  export default class pnpjs extends React.PureComponent<IpnpjsProps, IpnpjsS
113   private _getDemoItems = async(): Promise<void> => {
114     try {
115       const oldItems = this.state.items;
116       const webUrl = `${window.location.origin}/sites/Demos`;
117       // Optionally
118       // const webSP = spfi(webUrl).using(SPFx({ pageContext: this._pageContext }));
119       // const web = webSP.web;
120       const web = Web([this._sp.web, webUrl]);
121       let newItems: IFile[] = [];
122       if(web){
123         const response: IResponseItem[] = await web.lists
124         .getByTitle(this.LIBRARY_NAME)
125         .items
126         .select("Id", "Title", "FileLeafRef", "File/Length")
127         .expand("File")();
128
129         newItems = response.map((item: IResponseItem) => {
130           return {
131             Id: item.Id,
132             Title: item.Title || "Unknown",
133             Size: item.File?.Length || 0,
134             Name: item.FileLeafRef
135           };
136         });
137       }
138       const items = oldItems.concat(newItems);
139       this.setState({ items });
```

Behaviors

How to create your own

Behaviors

Recipes for extending PnPjs

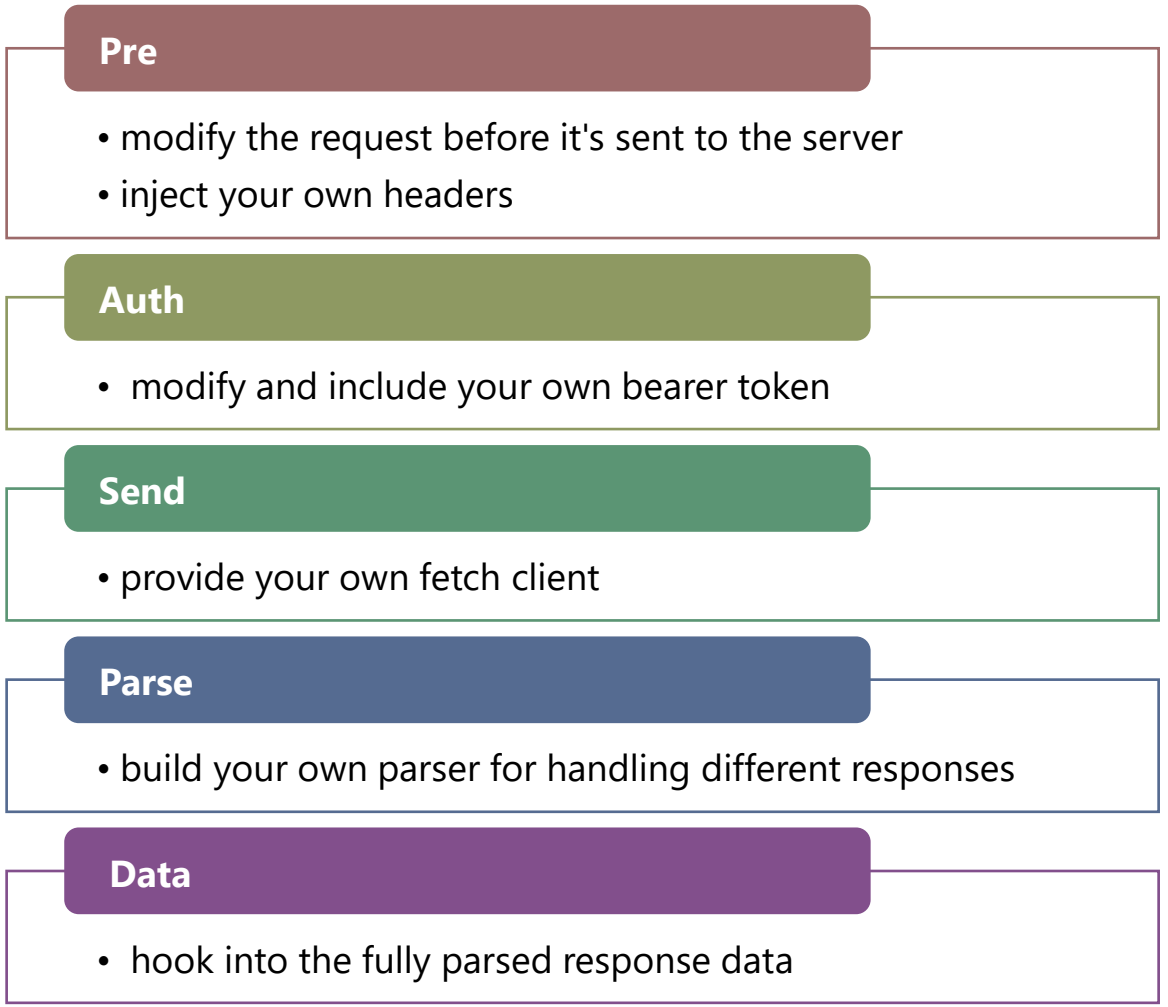
- Bearer Token
- Browser Fetch
- Browser Fetch with Retry
- Caching
- Inject Headers
- Node Fetch
- Parsers
  - Json
  - Text
  - Blob
  - Buffer
  - Header
- Resolvers

```
const sp = spfi().using(Caching());

const items = await sp.web.lists.getByTitle("Items").items();
```

Behaviors allow *hooking into* lifecycle events

Examples



Anatomy of a behavior

Clear – Remove active behaviors

Prepend – Apply before any other behaviors

Replace – Replace all behaviors with this

```
export function MyCustomParser(): TimelinePipe<Quer

    return (instance: Queryable) => {

        instance.on.parse.replace(async (url: URL, response: Response, result: any): Promise<[URL, Response, any]> =>

            if(response.ok && typeof result !== "undefined"){
                result = await response.json();
            }
            return [url, response, result];
        });

        return instance;
    };
}
```

Creating our own Behavior

```
function InjectHeaders(headers, prepend = false) {
    return (instance) => {
        instance.on.pre.prepend(async (url, init, result) => {
            init.headers = { ...init.headers, ...headers };
            return [url, init, result];
        });
        return instance;
    };
}
const sp = spfi().using(
    SPBrowser({ baseUrl: settings.testing.sp.url }),
    MSAL({ configuration: settings.testing.sp.msal.init, authParams: { scopes: settings.testing.sp.msal
    InjectHeaders({
    "Accept": "application/json;odata=verbose",
    "Beau-Custom": "myCustomHeader"
    })
);
const web = await sp.web();
```

NodeJS project

Azure Functions Demo

package.json

{} package.json > ...

```
{
  "name": "azfuncnode",
  "version": "1.0.0",
  "description": "",
  "type": "module",
  ▷ Debug
  "scripts": {
    "build": "tsc",
    "watch": "tsc -w",
    "clean": "rimraf dist",
    "prestart": "npm run clean && npm run build",
    "start": "func start",
    "test": "echo \"No tests yet...\""
  },
  "dependencies": {
    "@azure/functions": "4.4.0",
    "@pnp/azidjsclient": "4.0.0",
    "@pnp/graph": "4.0.0",
    "@pnp/nodejs": "4.0.0",
    "@pnp/sp": "4.0.0",
    "applicationinsights": "2.9.5"
  },
  "devDependencies": {
    "@types/node": "^18.x",
    "typescript": "^5.0.0",
    "rimraf": "^5.0.0"
  },
  "main": "dist/src/functions/*.js"
}
```

tsconfig.json

tsconfig.json > ...

```
1   {
2     "compilerOptions": {
3       "module": "ESNext",
4       "target": "ESNext",
5       "outDir": "dist",
6       "rootDir": ".",
7       "sourceMap": true,
8       "strict": false,
9       "moduleResolution": "Node",
10      "allowSyntheticDefaultImports": true,
11      "typeRoots": [
12        "./node_modules/@types"
13      ],
14    }
15  }
```

local.settings.json

{} local.settings.example.json > {} Values

```
{
  "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "UseDevelopmentStorage=true",
    "FUNCTIONS_WORKER_RUNTIME": "node",
    "AzureWebJobsFeatureFlags": "EnableWorkerIndexing",
    "APPLICATIONINSIGHTS_CONNECTION_STRING": "",
    "SiteUrl": "https://contoso.sharepoint.com/sites/pnpjs",
    "Tenant": "contoso",
    "ListGUID": "99999999-9999-9999-9999-999999999999",
    // Next 3 setting are only for local Azure Identity debugging
    "AZURE_CLIENT_ID": "99999999-9999-9999-9999-999999999999",
    "AZURE_TENANT_ID": "99999999-9999-9999-9999-999999999999",
    // path to .pem file
    "AZURE_CLIENT_CERTIFICATE_PATH":"c:\\cert.pem"
  }
}
```


pnpjsService.ts
import AppInsights from 'applicationinsights';
import { DefaultAzureCredential } from "@azure/identity";
import { MyItem } from './models.js';
import { spfi, SPFI } from "@pnp/sp";
import { AzureIdentity } from "@pnp/azidjsclient";
import { GraphDefault, SPDefault } from "@pnp/nodejs";
import "@pnp/sp/webs/index.js";
import "@pnp/sp/lists/index.js";
import "@pnp/sp/items/index.js";
import { graphfi, GraphFI } from '@pnp/graph';
export interface IPnpjsService {
Init: () => Promise<boolean>;
GetListItem: (id: string) => Promise<MyItem>;
}
export class PnpjsService implements IPnpjsService {
private LOG_SOURCE = "PnpjsService";
private _ready: boolean = false;
private _sp: SPFI = null;
private _graph: GraphFI = null;
public constructor() { }
public async Init(): Promise<boolean> {
public get ready(): boolean {
return this._ready;
public async GetListItem(id: string): Promise<MyItem> {
export const pnpjs: IPnpjsService = new PnpjsService();

src > common > TS pnpjsService.ts > PnpjsService > Init

pnpjsService.ts

```
18  export class PnpjsService implements IPnpjsService {
22    private _sp: SPFI = null;
23    private _graph: GraphFI = null;
24
25    public constructor() { }
26
27    public async Init(): Promise<boolean> {
28      let retVal = false;
29      try {
30
31        const credential = new DefaultAzureCredential();
32
33        this._sp = spfi(process.env.SiteUrl).using(SPDefault({}),
34          AzureIdentity(credential, [`https://${process.env.Tenant}.sharepoint.com/.default`], null));
35
36        this._graph = graphfi().using(GraphDefault({}), AzureIdentity(credential, [`https://graph.microsoft.com/.default`], null));
37
38        this._ready = true;
39        retVal = true;
40        AppInsights.defaultClient.trackTrace({
41          message: 'Init success',
42          properties: {
43            source: this.LOG_SOURCE,
44            method: "Init"
45          },
46          severity: AppInsights.Contracts.SeverityLevel.Verbose
47        });
48      } catch (err) {
49        AppInsights.defaultClient.trackException({
50          exception: err,
51          severity: AppInsights.Contracts.SeverityLevel.Critical
```

pnpjsService.ts

```
src > common > TS pnpjsService.ts > PnpjsService > Init
18  export class PnpjsService implements IPnpjsService {
57
58    public get ready(): boolean {
59      return this._ready;
60    }
61
62    public async GetListItem(id: string): Promise<MyItem> {
63      let retVal: MyItem = null;
64      try {
65        const item = await this._sp.web.lists.getById(process.env.ListGUID).items.getById(+id)();
66        retVal = { Id: item.Id, Title: item.Title, Description: item.Description };
67        AppInsights.defaultClient.trackTrace({
68          message: 'GetListItem success',
69          properties: {
70            source: this.LOG_SOURCE,
71            method: "GetListItem"
72          },
73          severity: AppInsights.Contracts.SeverityLevel.Verbose
74        });
75      } catch (err) {
76        AppInsights.defaultClient.trackException({
77          exception: err,
78          severity: AppInsights.Contracts.SeverityLevel.Critical,
79          properties: { source: this.LOG_SOURCE, method: "GetListItem" }
80        });
81      }
82      return retVal;
83    }
84  }
85
86  export const pnpjs: IPnpjsService = new PnpjsService();
87
```

httpTrigger.ts

```ts
import { app, HttpRequest, HttpResponseInit, InvocationContext } from "@azure/functions";
import AppInsights from 'applicationinsights';
import { pnpjs } from "../common/pnpjsService.js";

export async function httpTrigger(request: HttpRequest, context: InvocationContext): Promise<HttpResponseInit> {
    const LOG_SOURCE = "httpTrigger";
    // Initialize Application Insights, setting AutoDependencyCorrelation to true so that all logs for each run are correlated together
    in App Insights.
    AppInsights.setup(process.env.APPLICATIONINSIGHTS_CONNECTION_STRING).setAutoDependencyCorrelation(true);

    // start the client
    AppInsights.start();

    AppInsights.defaultClient.trackEvent({ ···
    });

    // Set the default response to a 200 with an empty body.
    let retVal: HttpResponseInit = { status: 200, body: "" };

    try { ···
    } catch (err) { ···
    }

    // Return the appropriate response to the requestor.
    return retVal;
};

app.http('httpTrigger', {
    methods: ['GET'],
    authLevel: 'anonymous',
    handler: httpTrigger
});
```

httpTrigger.ts

```
21
22  // Set the default response to a 200 with an empty body.
23  let retVal: HttpResponseInit = { status: 200, body: "" };
24
25  try {
26    // If the request is a GET
27    if (request.method == "GET") {
28      const id = request.query.get("id");
29      if (id != null) {
30        const ready = await pnpjs.Init();
31        if (ready) {
32          const result = await pnpjs.GetListItem(id);
33          if (result != null) {
34            retVal = { status: 200, body: JSON.stringify(result) };
35            AppInsights.defaultClient.trackTrace({···
42            });
43          } else {
44            retVal = { status: 400, body: "Item not found." };
45            AppInsights.defaultClient.trackTrace({···
52            });
53          }
54        }
55      } else {···
65      }
66    }
67  } catch (err) {···
73  }
74
75  // Return the appropriate response to the requestor.
76  return retVal;
77 };
```

What's changed in V4

Updates

- Bug fixes!
- Async Iterators
- File Labels
- Followed Sites Support
- Get All Sites()
- Mail & Mail Folders Operations
- Site & List Operations
- Drive Items (Files)

Removed/Deprecated

- Paged()
- Drive Item – Move
- Drive Item – Set Content
- SP Taxonomy
- Get-All

New Modules

- admin
- analytics
- app catalog
- compliance
- files
- list-item
- mail
- operations
- permissions
- places
- taxonomy
- to-do

Async Iterators – paging *improved*

Paging in v3

```
let items = await sp.web.lists.getByTitle("SPFx List").items.top(300).getPaged();

if(items.hasNext){
    items = await items.getNext();
    console.log(items.results.length);
}
```

V4 Paging w/ Async Iterators

```
for await (const items of sp.web.lists.getByTitle("SPFx List").items.top(300)) {
    console.log(items.length);
    // perform some logic
    //if logic is met, break out of loop
    break;
}
```

V4 Paging w/ Async Iterators

```
for await (const items of sp.web.lists.getByTitle("SPFx List").items.top(1)) {
    console.log(items.length);
    if (items[0].Id > 5) {
        break;
    }
}
```

V4 Updates – Removal of 'data'

```
let item = await sp.web.lists.getByTitle("SPFx List").items.add({
    "Title": "Test Item"
});

console.log(item.Title);

let itemQueryable = await sp.web.lists.getByTitle("SPFx List").items.getById(item.data.Id);
```

Questions

Resources

Repository: https://github.com/pnp/pnpjs

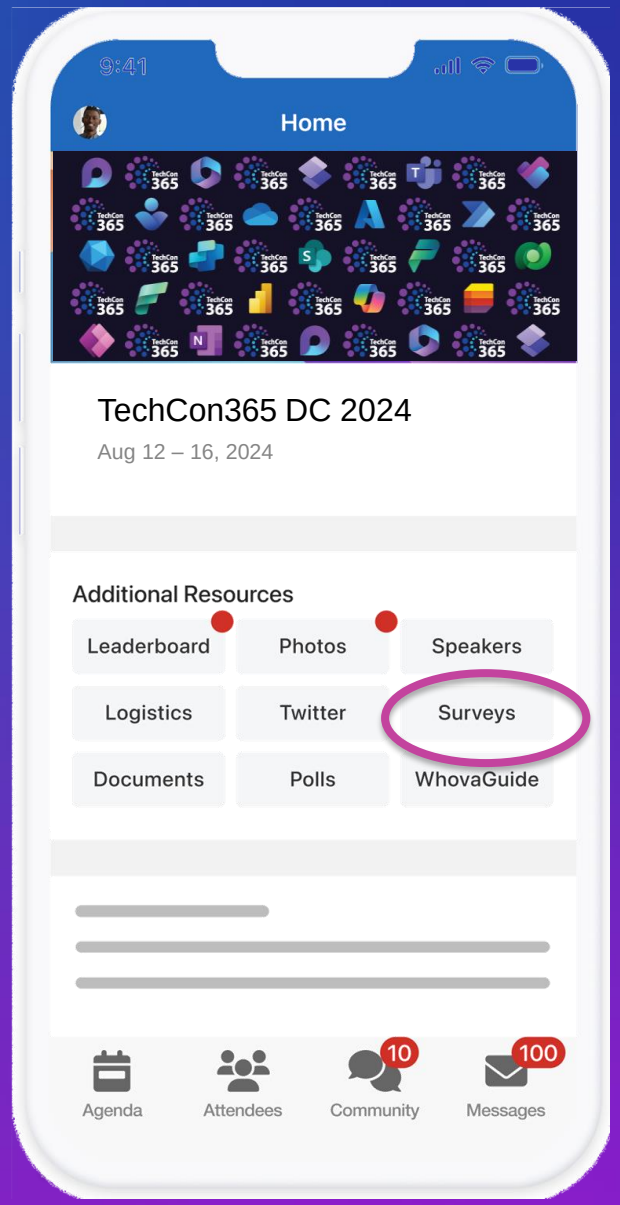
Documentation: https://pnp.github.io/pnpjs/

Today's Sample: https://github.com/pnp/pnpjs/tree/version-4/samples/spfx-react-components

How was the session?

Search for WHOVA in the App Store or Google Play

Fill out the Surveys in the **TechCon 365 DC Event App** and be eligible to win **PRIZES!**



TechCon 365

techcon365.com/DC