#SPSBE

APIs, APIs Everywhere!

Sébastien Levert | @sebastienlevert

BIWUG

SharePoint Saturday Belgium 2018

Thanks to our sponsors!

Platinum

Gold

Silver

SharePint

Community

Hi! I'm Seb!

Microsoft® Most Valuable Professional

@sebastienlevert | http://sebastienlevert.com | Product Evangelist & Partner Manager at Valo

Agenda

Valo

SharePoint REST APIs

Microsoft Graph & Custom APIs

PnP JS Core

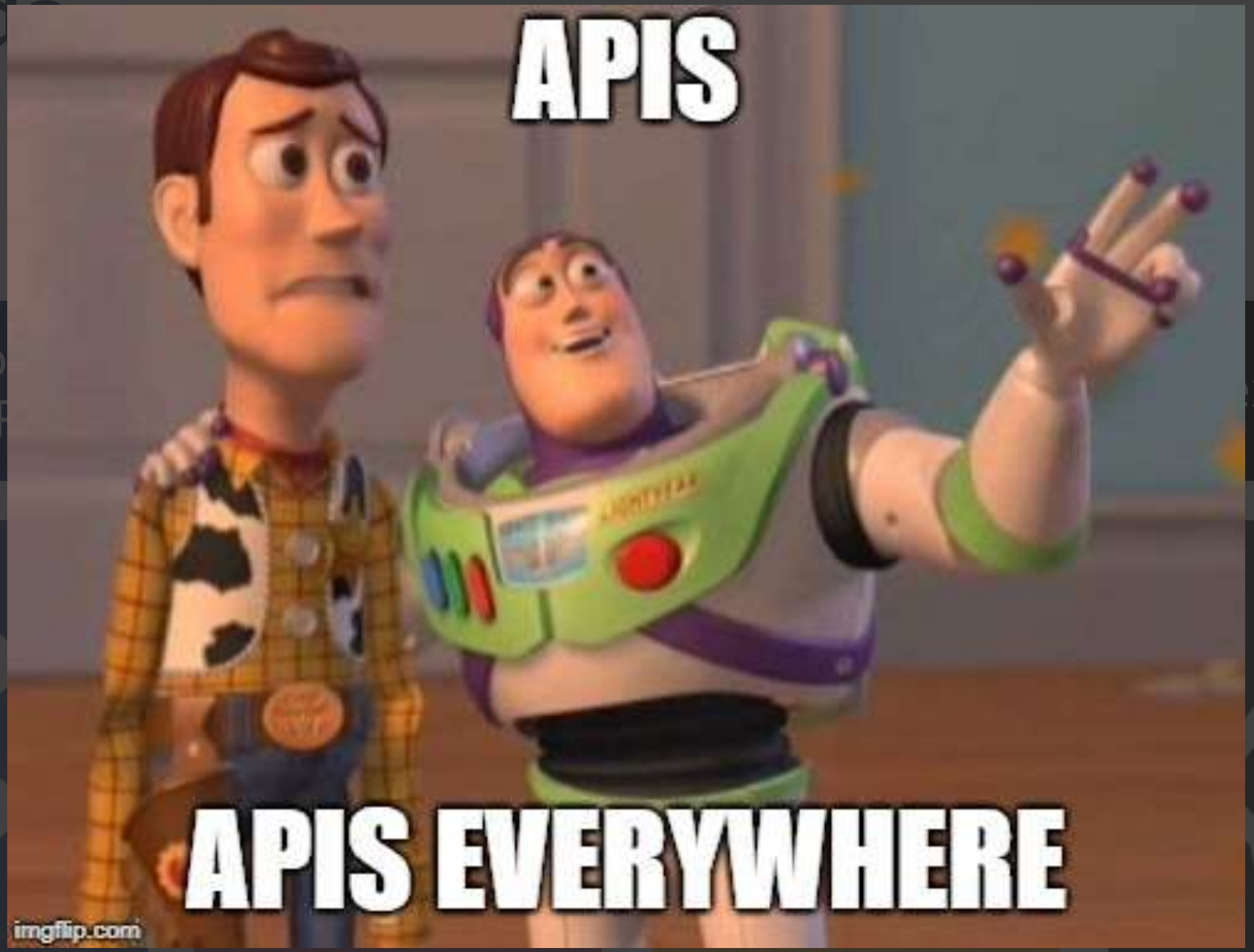
Next Steps

Agend

Valo

SharePo
REST AP

xt Steps

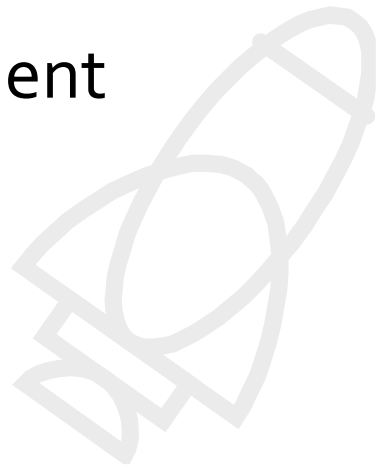


SharePoint REST APIs

Our Scenario

Valo

- Building a SharePoint Framework webpart that connects to a SharePoint list to play with its data
- Using a single Interface to define our Data Access services to enable easy on-the-fly switch of data sources
- Mocking Service for swapping and local Workbench development



Data Service Architecture

```typescript
import { IHelpDeskItem } from "./../models/IHelpDeskItem";
import { WebPartContext } from "@microsoft/sp-webpart-base";

export default interface IDataService {
  getTitle(): string;
  isConfigured(): boolean;
  getItems(context: WebPartContext): Promise<IHelpDeskItem[]>;
  addItem(context: WebPartContext, item: IHelpDeskItem): Promise<void>;
  updateItem(context: WebPartContext, item: IHelpDeskItem): Promise<void>;
  deleteItem(context: WebPartContext, item: IHelpDeskItem): Promise<void>;
}

export default class SharePointDataService implements IDataService {
  //…
}
```

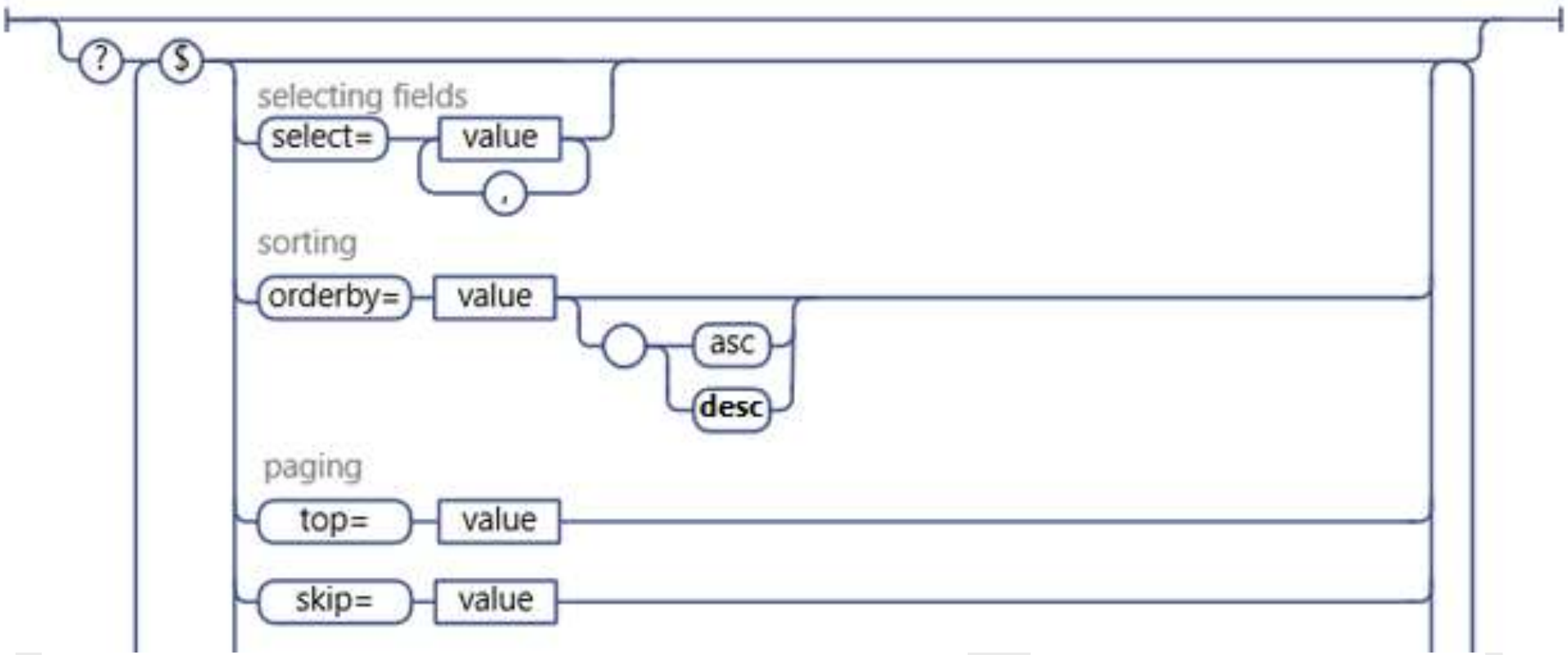
Using the SharePoint REST APIs?

Valo

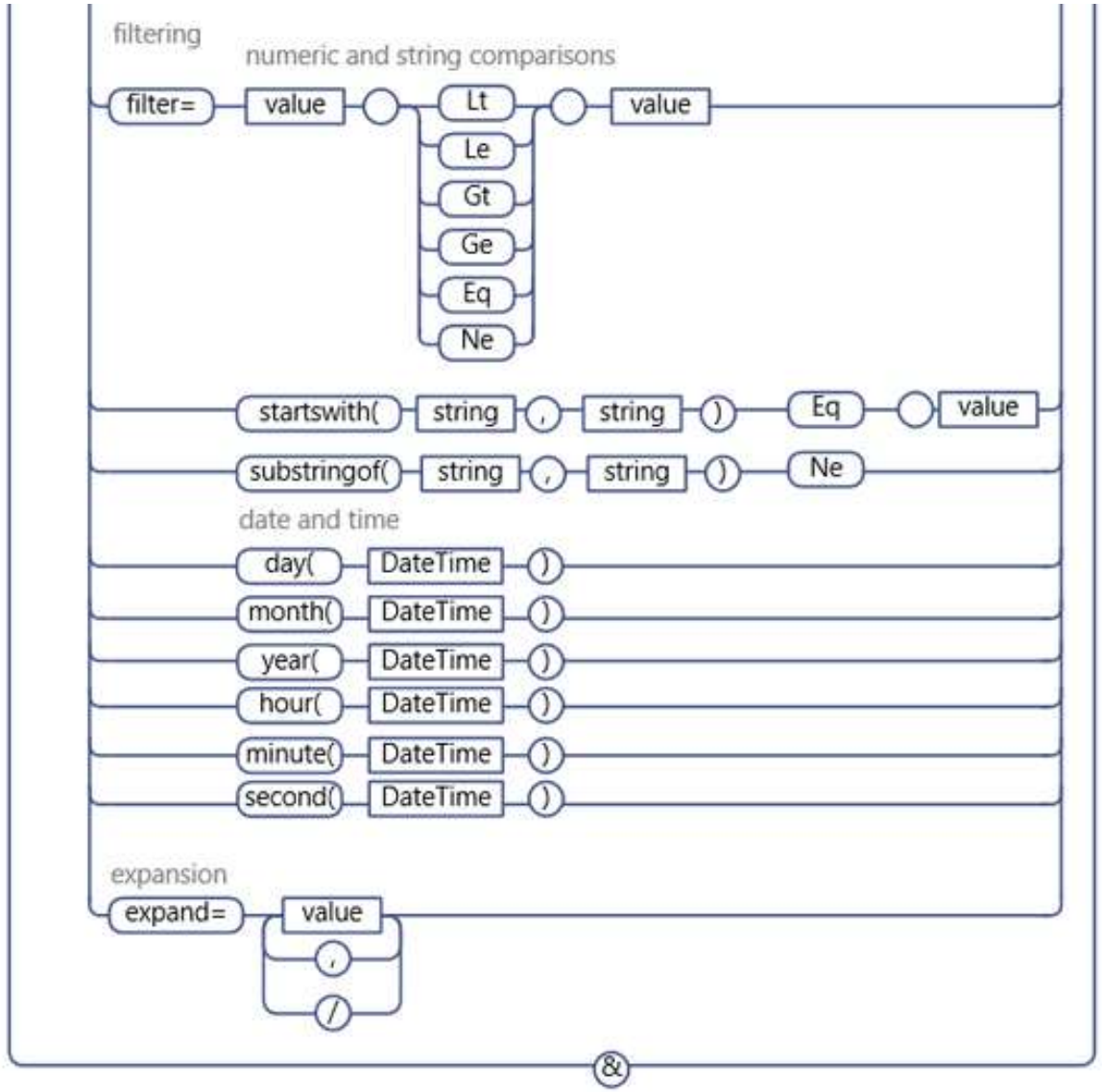
- Enable almost all your CRUD scenarios in the solutions you are building on SharePoint Online and SharePoint On-Premises
- When called from a SharePoint context, no authentication required as it's all cookie based
- It follows the OData standards, making it easy to query your content

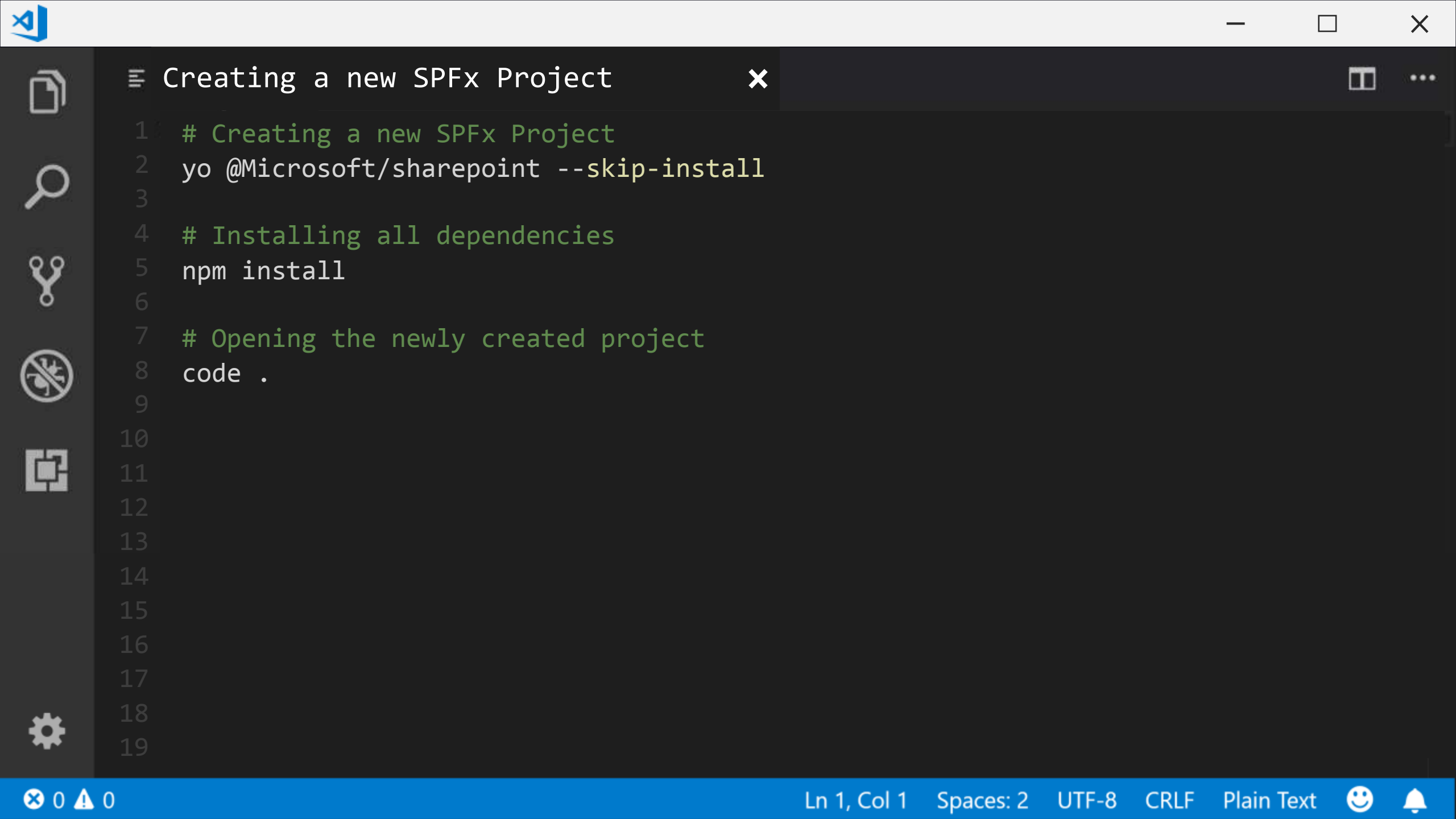
OData URI at a glance

Valo



OData URI at a glance




Creating a new SPFx Project
Creating a new SPFx Project
yo @Microsoft/sharepoint --skip-install
Installing all dependencies
npm install
Opening the newly created project
code .
0 0
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
Plain Text

Retrieving Data

```
// Getting all the sessions of the specified list using SharePoint REST APIs
public getItems(context: WebPartContext): Promise<IHelpDeskItem[]> {
  return new Promise<IHelpDeskItem[]>((resolve, reject) => {
    context.spHttpClient.get(
      `${absoluteUrl}/_api/web/lists/GetById('${this._listId}')/items` +
      `?$select=*,HelpDeskAssignedTo/Title&$expand=HelpDeskAssignedTo`,
      SPHttpClient.configurations.v1)
    .then(res => res.json())
    .then(res => {
      let helpDeskItems:IHelpDeskItem[] = [];

        for(let helpDeskListItem of res.value) {
          helpDeskItems.push(this.buildHelpDeskItem(helpDeskListItem));
        }

        resolve(helpDeskItems);
     }).catch(err => console.log(err));
  });
}
```

0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text

Creating Data

```
// Creating a new session in the specified list
public addItem(item: IHelpDeskItem): Promise<void> {
  return new Promise<void>((resolve, reject) => {
 //…
    return this._webPartContext.spHttpClient.post(
      `${currentWebUrl}/_api/web/lists/GetById('${this._listId}')/items`,
      SPHttpClient.configurations.v1, {
        headers: { "Accept": "application/json;odata=nometadata",
                   "Content-type": "application/json;odata=verbose",
                   "odata-version": "" },
        body: body
      });
    }).then((response: SPHttpClientResponse): Promise<any> => {
      return response.json();
    }).then((item: any): void => {
      resolve();
    });
  });
}
```

Deleting Data

```
// Deleting a specific item from the specified list
public deleteItem(id: number): Promise<void> {
  return new Promise<void>((resolve, reject) => {
    if (!window.confirm(`Are you sure?`)) { return; }

    return this._webPartContext.spHttpClient.post(
      `${currentWebUrl}/_api/web/lists/GetById('${this._listId}')/items(${id})`,
      SPHttpClient.configurations.v1, {
        headers: { "Accept": "application/json;odata=nometadata",
                   "Content-type": "application/json;odata=verbose",
                   "odata-version": "",
                   "IF-MATCH": "*",
                   "X-HTTP-Method": "DELETE" }
    }).then((response: SPHttpClientResponse): void => {
      resolve();
    });
  });
}
```

Using SharePoint Search

Valo

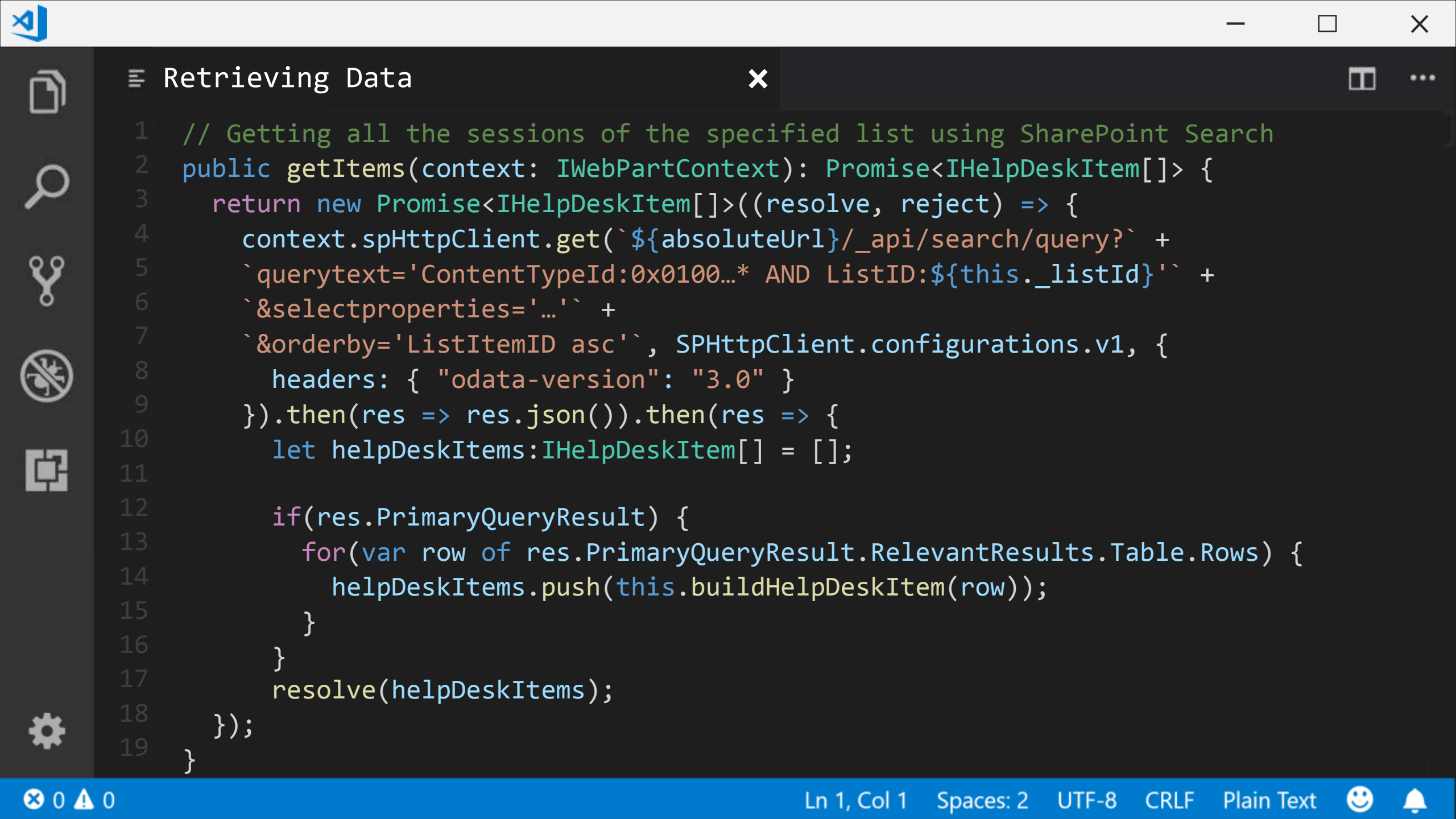
- Using search allows you to query content in multiple lists or multiple sites or site collections
- Uses a totally other query language (KQL or Keyword Query Language)
- Very performant and optimized for fetching a lot of data, but has a 15 minutes-ish delay in terms of data freshness
- Does not support any data modification

KQL Crash Course

Valo

- See Mickael Svenson blog series “SharePoint Search Queries Explained”

- https://www.techmikael.com/2014/03/sharepoint-search-queries-explained.html


Retrieving Data
// Getting all the sessions of the specified list using SharePoint Search
public getItems(context: IWebPartContext): Promise<IHelpDeskItem[]> {
return new Promise<IHelpDeskItem[]>((resolve, reject) => {
context.spHttpClient.get(`${absoluteUrl}/_api/search/query?` +
`querytext='ContentTypeId:0x0100…* AND ListID:${this._listId}'` +
`&selectproperties='…'` +
`&orderby='ListItemID asc'`, SPHttpClient.configurations.v1, {
headers: { "odata-version": "3.0" }
}).then(res => res.json()).then(res => {
let helpDeskItems:IHelpDeskItem[] = [];
if(res.PrimaryQueryResult) {
for(var row of res.PrimaryQueryResult.RelevantResults.Table.Rows) {
helpDeskItems.push(this.buildHelpDeskItem(row));
}
}
resolve(helpDeskItems);
});
}
0 0
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
Plain Text

Notes on legacy APIs support

Valo

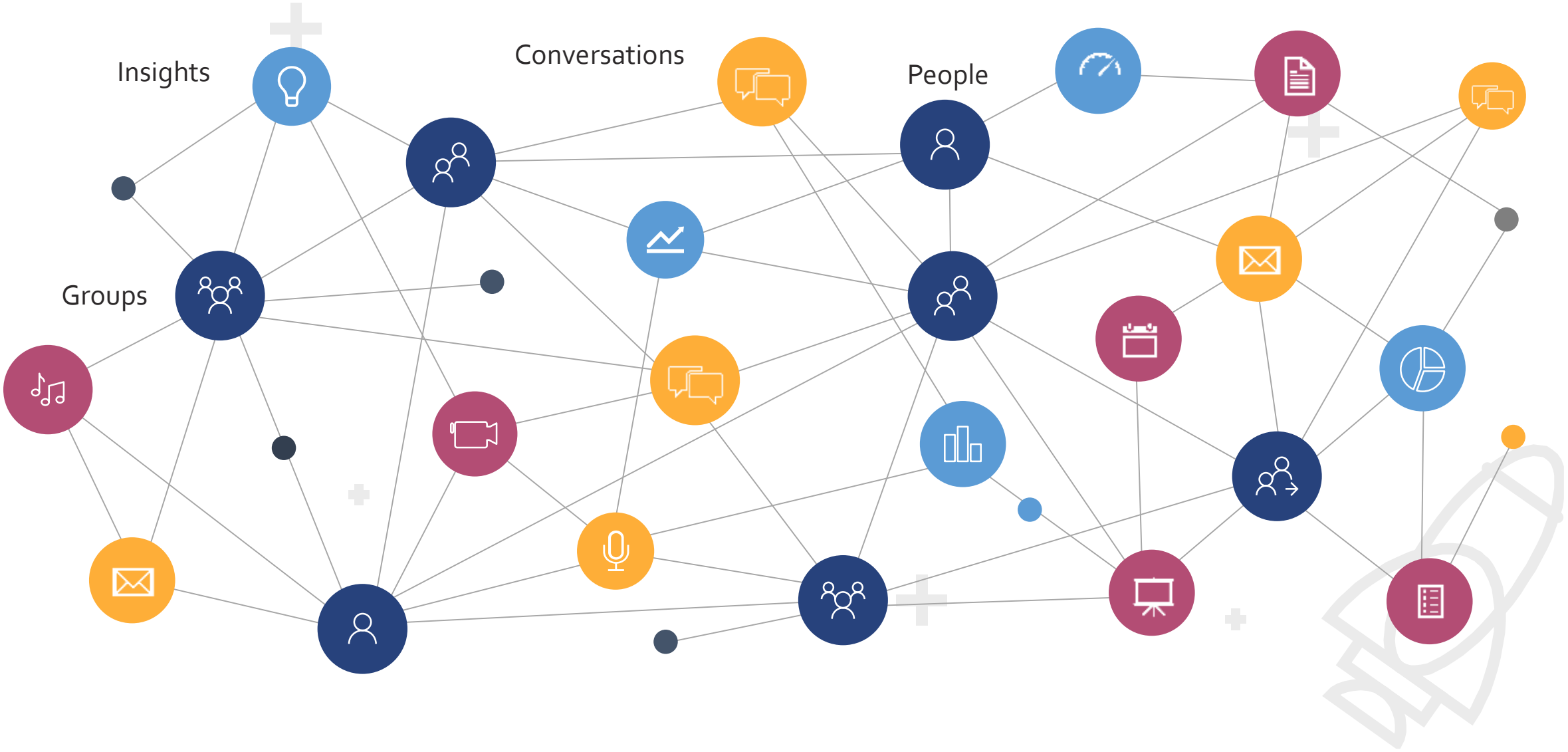
- SharePoint APIs cover a wide-range of options, but not everything
- You “might” have to revert to JSOM for some scenarios (Managed Metadata, etc.)
- Or even to the ASMX Web Services for more specific scenarios (Recurring Events in Calendars, etc.)
- The SharePoint Framework supports those scenarios, but will require some extra work

Valo

Microsoft Graph and Custom APIs

What is the Microsoft Graph?

Valo



Microsoft Graph is all about you

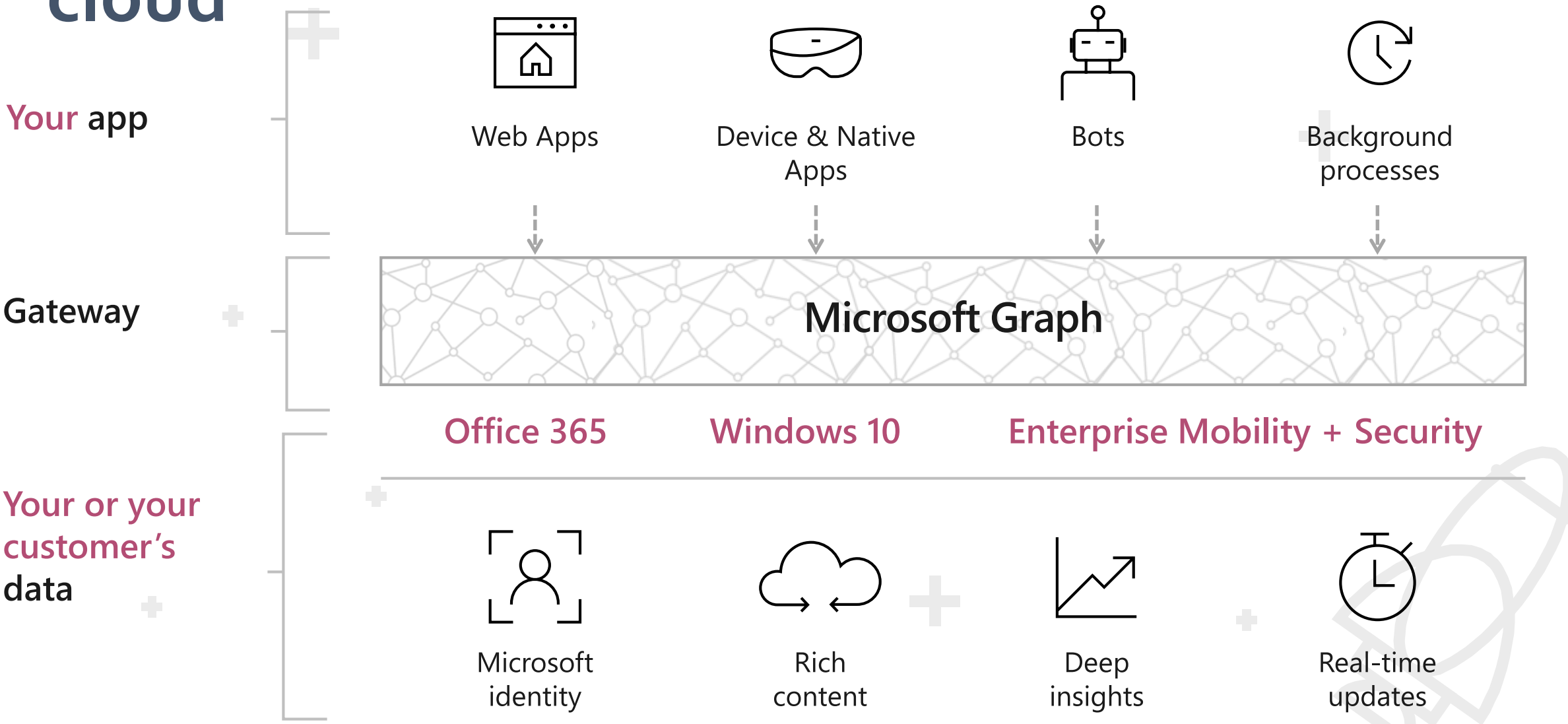
Valo

If you or your customers are part of the millions of users that are using Microsoft cloud services, then Microsoft Graph is the fabric of all your data

It all starts with /me

Gateway to your data in the Microsoft cloud

Valo



Microsoft Graph

Valo

https://graph.microsoft.com

ALL

your data across Microsoft 365

- Office 365
- Windows 10
- EMS

ALL

types of users

- Corporate (@contoso.com)
- Consumer (@outlook.com)

ONE

way to access it

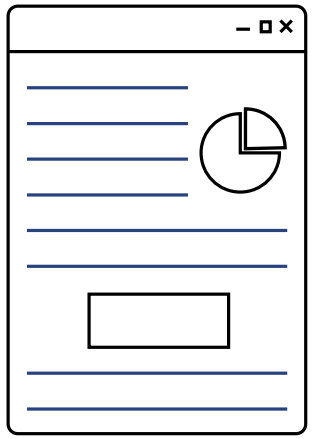
- One endpoint
- One auth key
- One set of docs
- One SDK

Microsoft 365 Platform

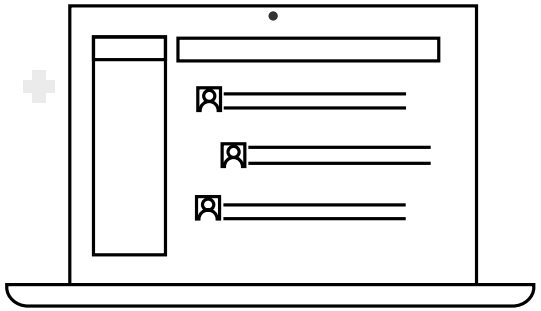
Valo

Extend Microsoft 365 experiences

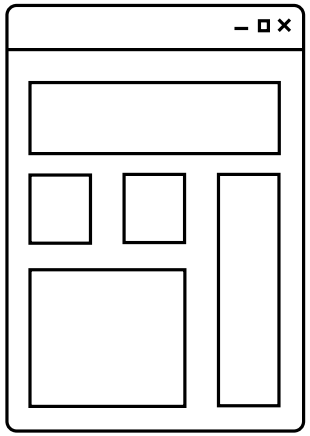
Documents



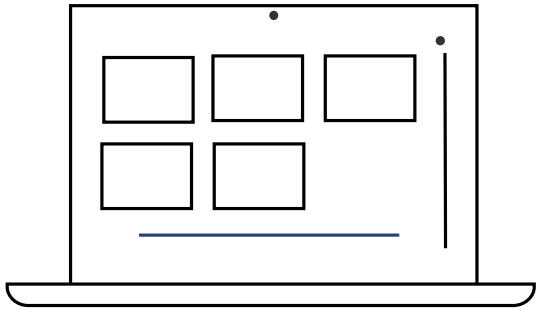
Conversations



Pages

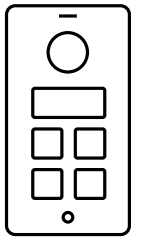


Timeline



Build your experience

web, device,
and service apps



iOS/Android/Windows/Web

Microsoft Graph

Options for the Microsoft Graph

Valo

- Using the built-in MsGraphClient class that takes care of the entire authentication flow and token management for you

- Using a custom implementation using ADAL or MSAL

Are there any limitations?

Valo

- Every scope will have to be defined beforehand and could open some security challenges

- Authentication challenges could bring a broken experience to your users depending on the platform and browser used

Enabling Graph in your solution

```
// In your package-solution.json
{
  // …
  "webApiPermissionRequests": [
    {
      "resource": "Microsoft Graph",
      "scope": "Sites.ReadWrite.All"
    },
    // …
  ]
  // …
}
```

Retrieving Data

```
// Getting all the sessions of the specified list using the Microsoft Graph
public getItems(context: WebPartContext): Promise<IHelpDeskItem[]> {
  let graphUrl: string = `https://graph.microsoft.com/v1.0` +
                         `/sites/${this.getCurrentSiteCollectionGraphId()}` +
                         `/lists/${this._listId}` +
                         `/items?expand=fields(${this.getFieldsToExpand()})`;

  return new Promise<IHelpDeskItem[]>((resolve, reject) => {
    this._client.api(graphUrl).get((error, response: any) => {
      let helpDeskItems:IHelpDeskItem[] = [];

      for(let helpDeskListItem of response.value) {
        helpDeskItems.push(this.buildHelpDeskItem(helpDeskListItem));
      }

      resolve(helpDeskItems);
    });
  });
}
```

Creating Data

```
// Creating a new item in the specified list
public addItem(item: IHelpDeskItem): Promise<void> {
  let graphUrl: string = `https://graph.microsoft.com/v1.0` +
                         `/sites/${this.getCurrentSiteCollectionGraphId()}` +
                         `/lists/${this._listId}` +
                         `/items`;
  return new Promise<void>((resolve, reject) => {
    const body: any = { "fields" : {
      "Title": item.title,
      "HelpDeskDescription": item.description,
      "HelpDeskLevel": item.level
    }};

    this._client.api(graphUrl).post(body, (error, response: any) => {
      resolve();
    });
  });
}
```

Deleting Data

```
// Deleting the specified item form the specified list
public deleteItem(id: number): Promise<void> {
  let graphUrl: string = `https://graph.microsoft.com/v1.0` +
                         `/sites/${this.getCurrentSiteCollectionGraphId()}` +
                         `/lists/${this._listId}` +
                         `/items/${id}`;

  return new Promise<void>((resolve, reject) => {
    this._client.api(graphUrl).delete((error, response: any) => {
      resolve();
    });
  });
}
```

Custom APIs

Valo

- Using the built-in AadGraphClient class that takes care of the entire authentication flow and token management for you
- Using a custom implementation using ADAL or MSAL
- You can use user impersonation (or not) to query SharePoint content based on user permissions

Retrieving Data

```
// Getting all the sessions of the specified list using the AAD Client
public getItems(context: WebPartContext): Promise<IHelpDeskItem[]> {
  return new Promise<IHelpDeskItem[]>((resolve, reject) => {
    let apiUrl: string = "<url>/api/GetHelpDeskItems";
    this._client
      .get(apiUrl, AadHttpClient.configurations.v1)
      .then((res: HttpClientResponse): Promise<any> => {
        return res.json();
      }).then((res: any): void => {
        let helpDeskItems:IHelpDeskItem[] = [];
        for(let helpDeskListItem of res) {
          helpDeskItems.push(this.buildHelpDeskItem(helpDeskListItem));
        }

        resolve(helpDeskItems);
      });
    });
  }
}
```

0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text

Patterns & Practices JS Core

What is PnP JS Core?

Valo

PnPJS is a fluent JavaScript API for consuming SharePoint and Office 365 REST APIs in a type-safe way. You can use it with SharePoint Framework, Nodejs, or JavaScript projects. This an open source initiative that complements existing SDKs provided by Microsoft offering developers another way to consume information from SharePoint and Office 365.

https://github.com/pnp/pnpjs

Benefits of PnP JS Core

Valo

- Type safe so you get your errors while you code and not when you execute and test
- Works on all versions of SharePoint (On-Premises, Online, etc.)
- Offers built-in caching mechanisms
- Heavily used in the SharePoint Development Community

Sample – Setup PnP JS for SharePoint

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

// ...

public onInit(): Promise<void> {

  return super.onInit().then(_ => {

    // other init code may be present

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

// ...
```

0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text

Retrieving Data

```
// Getting all the sessions of the specified list using PnP JS Core
public getItems(context: WebPartContext): Promise<IHelpDeskItem[]> {
  return new Promise<IHelpDeskItem[]>((resolve, reject) => {

    sp.web.lists.getById(this._listId).items
      .select("*", "HelpDeskAssignedTo/Title")
      .expand("HelpDeskAssignedTo").getAll().then((sessionItems: any[]) => {
        let helpDeskItems:IHelpDeskItem[] = [];

        for(let helpDeskListItem of sessionItems) {
          helpDeskItems.push(this.buildHelpDeskItem(helpDeskListItem));
        }

        resolve(helpDeskItems);
      });
  });
}
```


Valo
Next Steps

Resources

Valo

- https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/guidance/connect-to-sharepoint-using-jsom
- https://www.techmikael.com/2014/03/sharepoint-search-queries-explained.html
- https://github.com/pnp/pnpjs
- https://github.com/sebastienlevert/apis-apis-everywhere

Share your experience

Valo

- Use hashtags to share your experience
  - #Office365Dev
  - #MicrosoftGraph
  - #SPFx

- Log issues & questions to the GitHub Repositories

Thanks!

Microsoft® Most Valuable Professional

@sebastienlevert | http://sebastienlevert.com | Product Evangelist & Partner Manager at Valo

#SPSBE

Please rate this session!

http://spsbe.be


#SPSBE
THANK
YOU
BIWUG
SharePoint Saturday Belgium 2018