SPFX Extension

SPFx - extend user experience thru notifs, toolbars, list data views.

3 extension types:

Application customizer - scripts, accesses html, custom rendering

Field customizer - modified view for fields within a list

Command sets - new actions, behaviors

Tenant-wide Deployment of SharePoint Framework Extensions

Ez option for tenant app catalog manager

SharePoint Framework extensions are supported with modern experiences regardless of the actual site template used when content site was created.

SPFx component be available across the tenant using skipFeatureDeployment in json

App catalog site collection is used to deploy SharePoint add-ins and SharePoint Framework components to the tenant.

**Tenant Wide Extensions** list, which can be used to manage SharePoint Framework Extension activations in the tenant. Found in App Catalog

**Build**

Need the id in the manifest json

Uses onInit() instead of render

You cannot test ur extension on local workbench but u don’t have to deploy on the SP tenant app catalog

Serve.json? used to update page url to match ur sp site and test the extension

Need to be an actual page to test

Deployment

Elements xml – used by default when we explicitly install this sol to an existing site

Custom action in element – automatically added?

Clientsideextension.applicationcustomizer – defines that the sol is a applicationcustomizer

Clientsidecomponentsid = need to match the id in manifest.json

Need to update this in elements.xml?

ClientSideComponentProperties="{&quot;Top&quot;:&quot;Top area of the page&quot;,&quot;Bottom&quot;:&quot;Bottom area in the page&quot;}">

ClientsideInstance - used with [Tenant Wide deployment of SharePoint Framework](https://docs.microsoft.com/en-us/sharepoint/dev/spfx/extensions/basics/tenant-wide-deployment-extensions) so the sol can be deployed across all sites in tenant. Doesn’t matter if u will deploy ur sol on site level only. If u will not use this, delete it in config package-sol.json on element manifest.

Gulp serve nobrowser – serve the files which will be soon requested by the browser

TEKNIK WHEN RUNNING IN LOCAL?

FIELD customizer extension – used to change the render of the field

DEBUGGING!!!

Run gulp serve first then copy the remaining lines after ur url site to copy the id. Start debugging using hosted workbench. Paste the copied lines to the new instance of browser created from running debug mode. Take note: Launch.json url under hosted workbench needs to be ur site url

Serve.json – urls also need to be ur site url

SPFx API

SPHttpClient – used to connect to REST APIs

QUESTION ODATA?

Some classes for executing web requests: **SPHttpClient** (includes auth cookies) and **HttpClient**

PnPjs – alternative way of requesting APIs. Open src JS library for communicating with SP and office 365

**MSGraphClient** – simplifies connecting to MS Graph inside SPFx sol

The **MSGraphClient** is available only in projects built using SharePoint Framework v1.6.0 and later

Microsoft Graph is a Microsoft developer platform that connects multiple services and devices. For integration?? AN API. Useful with Azure AD when trying to identify users with their particular groups

The OAuth 2.0 authorization code grant can be used in apps that are installed on a device to gain access to protected resources, such as web APIs. Client-side applications that are incapable of storing a secret, such as SharePoint Framework solutions, use a specific type of OAuth flow named [OAuth implicit flow](https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-dev-understanding-oauth2-implicit-grant).

**AadHttpClient -** the low-level type used to access any Azure AD-secured REST API

**DEBUGGING**

Needs to install debugger for chrome – can be installed using npm or vs code

Launch.json – contains the configurations for debugging. Found in .vscode folder?

* Also gives u the option to add ur own tenant so that it would be available in the dropdown found in the debugging icon on the right

Breakpoints exists

Press F5 to start debugging or click debugging icon on the left then press play icon

Click debugging via toolbar run on top

Gulp serve –nobrowser = mean it won’t start with the browser yet but will just compile?

Testing on modern SP pages