Develop SharePoint Framework Web Part

1. Open command prompt. Create a directory for SPFx solution.

1. md spfx-sp2019-loaddata

2. Navigate to above created directory.

1. cd spfx-sp2019- loaddata

3. Run Yeoman SharePoint Generator to create the solution.

1. yo @microsoft/sharepoint

4. Yeoman generator will present you with the wizard by asking questions about the solution to be created.

**A screenshot of a computer program

Description automatically generated**

5. Yeoman generator will perform scaffolding process to generate the solution. The scaffolding process will take significant amount of time.

A computer screen with text on it

Description automatically generated

6.In the command prompt type below command to open the solution in code editor of your choice.

code .

It will open Visual Studio code

Go to src\webparts\LoadSharePointDataWebPart.ts

1. Import SPHttpClient

*import {SPHttpClient, SPHttpClientResponse, ISPHttpClientOptions } from '@microsoft/sp-http';*

1. Add Checkbox property pane

import {

IPropertyPaneConfiguration,

PropertyPaneTextField,

**PropertyPaneCheckbox**

} from '@microsoft/sp-property-pane';

1. Add the WebPartProps

export interface ILoadSharePointDataWebPartProps {

listname: string; (Here rename description to Listname)

checkbox1:boolean;

}

1. Add interfaces to maintain the SharePoint list data

export interface ISPListItems{

value: ISPListItem[];

}

export interface ISPListItem

{

Title: string;

Id: string;

Created: string;

}

1. Open LoadSharePointDataWebPart.module.scss file and add below css classes

.Table {

display: table;

}

.Title {

display: table-caption;

text-align: center;

font-weight: bold;

font-size: larger;

}

.Heading {

display: table-row;

font-weight: bold;

text-align: center;

}

.row {

display: table-row;

}

.cell {

display: table-cell;

border: solid;

border-width: thin;

padding-left: 5px;

padding-right: 5px;

}

1. Go to WebPart class export default class LoadSharePointDataWebPart extends BaseClientSideWebPart

Declare below variables

private listName: string = "";

private checkbox1Property: string = "Created";

1. In the render() method modify the html

<div class="${ styles.loadSharePointData }">

<div class="${styles.Table}">

<div class = "${styles.Heading}">

<div class="${styles.cell}">Title</div>

</div>

</div>

</div>

1. Get Listname from Propertypane, add below code in render method

this.listName = this.properties.listname;

1. Get Data from SharePoint list, call below method from render method

this.loadData();

1. Then add below methods, next to render method#

private loadData(): void{

let url:string = this.context.pageContext.web.absoluteUrl + "/\_api/web/lists/getbytitle('" + this.listName +"')/items?select=Title";

if(this.properties.checkbox1)

{

url += ",Created";

this.domElement.querySelector("." + styles.Heading).innerHTML += `<div class="${styles.cell}">Created</div>`;

}

this.GetListData(url).then((response)=> {

this.RenderListData(response.value);

});

}

private GetListData(url:string):Promise<ISPListItems>{

return this.context.spHttpClient.get(url,SPHttpClient.configurations.v1)

.then((response:SPHttpClientResponse) =>{

return response.json();

});

}

private RenderListData(listItems:ISPListItem[]):void{

let strHtml:string = "";

if(listItems)

{

listItems.forEach((listItem:ISPListItem)=>{

let itemcreatedstr:string = listItem.Created;

strHtml += `<div class="${styles.row}">`;

strHtml += `<div class="${styles.cell}"><p>${listItem.Title}</p></div>`;

if(this.properties.checkbox1)

{

strHtml += `<div class="${styles.cell}"><p>${listItem.Created}</p></div>`;

}

strHtml += `</div>`;

});

}

this.domElement.querySelector("." + styles.Table).innerHTML += strHtml;

}

1. Update the Property pane controls

Replace the groupFields

groupFields: [

PropertyPaneTextField('listname', {

label: "List Name"

}),

PropertyPaneCheckbox('checkbox1',{

text:this.checkbox1Property

})

]

Open Command Prompt

**gulp Serve**

add the webpart and verify no errors

deploy and Test using your sharepoint site (follow previous lab 30 app catalog session

**gulp budle –ship**

**gulp package-solution –ship**

This will create the solution package (sppkg) in sharepoint\solution folder.

1. Open SharePoint App Catalog site.

2. From left menu, click Apps for SharePoint.

3. Click Upload to add SPFx package from “\sharepoint\solution” location.

4. Click OK.

A screenshot of a computer

Description automatically generated

5. Check the check box and Click Deploy.

Test the Webpart

Go to a Modern site and add your webpart

A screenshot of a computer

Description automatically generated

A blue rectangle with white border

Description automatically generated

Click edit icon in the left

Create a new list called Country and some country names

A screenshot of a computer

Description automatically generated

Then add any list name in the right side “List Name”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated