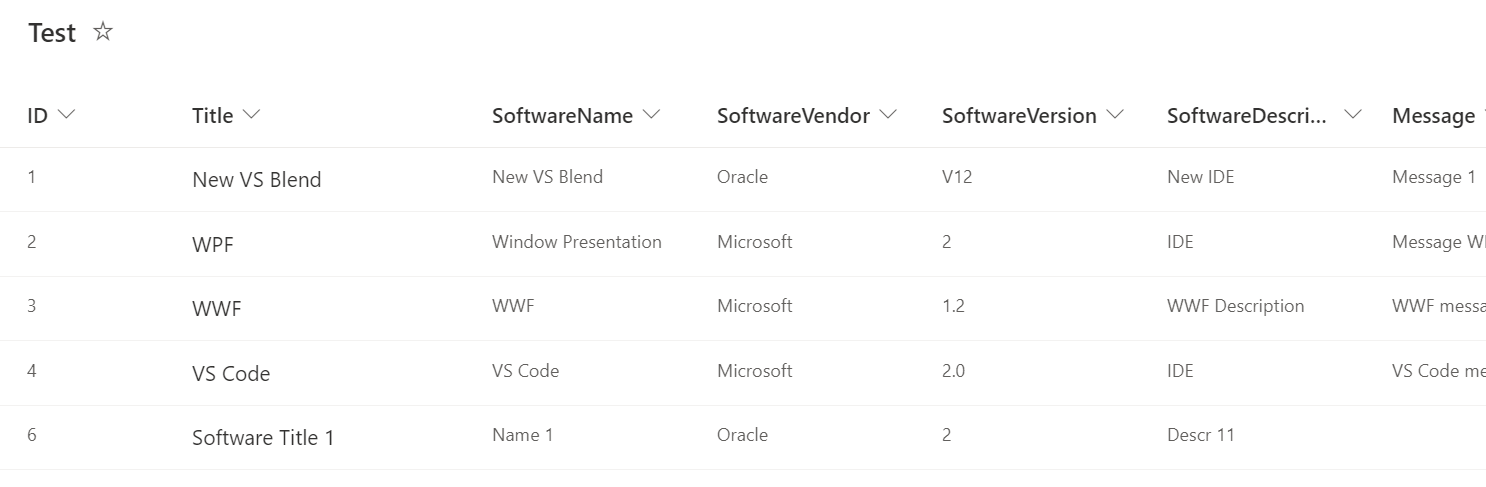
**SharePoint Framework (SPFx) CRUD operation using PNP JS with React Framework.**

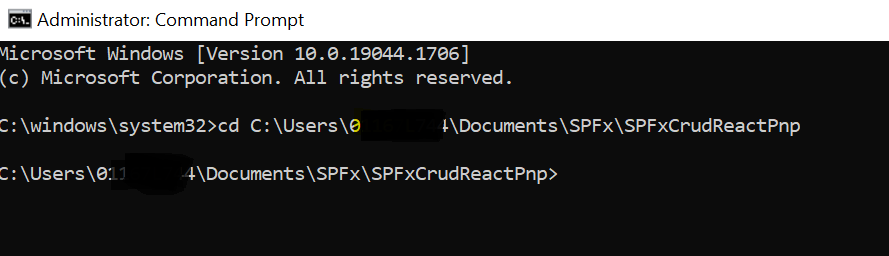
Create a list with Name **Test**.



**Step 1:**Create a folder with the name SPFxCrudReactPnp on your local drive.

C:\Users\ABC\Documents\SPFx\**SPFxCrudReactPnp**

**Step 2:**Open the location in the command prompt.

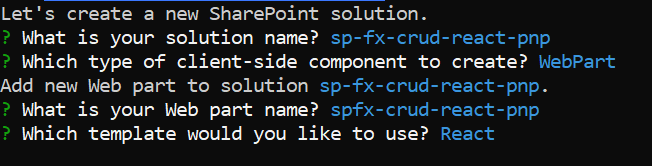


**Step 3:** Scaffold the SPFx solution using Yeoman Generator

[?](https://www.office365notes.com/2021/04/spfx-crud-operations-using-pnp-pnpsp.html)

|  |  |
| --- | --- |
|  | yo @microsoft/sharepoint |

**Step 4:**Give the webpart name and other details as shown below



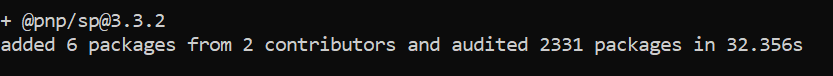
**Step 5:**Open the solution in VS Code. You can use the "Code ." command in the command prompt directly to open the solution in VS Code.

**Step 6:** In this SharePoint Framework webpart, we are using PNP JS for performing crud operations. Therefore we need to install the [PNP](https://pnp.github.io/pnpjs/sp/) in our solution. To install the PNP JS, run the below command in the command prompt. Note: This article is using version 2 of pnp/sp. Please check the note version at bottom of this document.

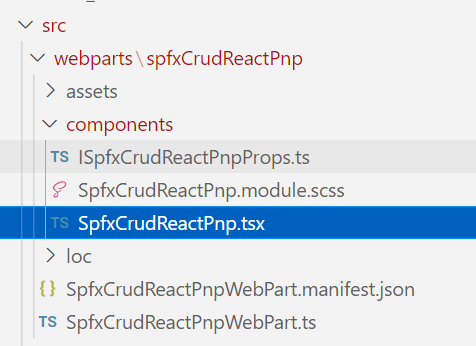
|  |  |
| --- | --- |
|  | npm i @pnp/sp |



**Step 7:** Once the PNP module is installed, you can see the message on the command prompt for the same



**Step 8:**We are going to write the entire logic for CRUD operations in **SpfxCrudReactPnp.tsx** file. For that, we need the context of SharePoint in .tsx file.



**Step 9:** In PNP it is very easy to set up the SPFx context. Just import the sp module and update the Oninit method in **SpfxCrudReactPnpWebPart.ts**as shown below:

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

*protected onInit(): Promise<void> {*

*this.\_environmentMessage = this.\_getEnvironmentMessage();*

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

*sp.setup({*

*spfxContext: this.context,*

*});*

*});*

*}*

After adding the above code, your file looks as below:



**Step 10:** Update the **SpfxCrudReactPnp.tsx**file with the below JSX (React way of writing HTML and javascript).

*<div className={styles.spfxCrudReactPnp}>*

*<div className={styles.container}>*

*<div className={styles.row}>*

*<div className={styles.column}>*

*<div className={styles.itemField}>*

*<div className={styles.fieldLabel}>ID:</div>*

*<input type="text" id="itemId"></input>*

*</div>*

*<div className={styles.itemField}>*

*<div className={styles.fieldLabel}>Software Title</div>*

*<input type="text" id="title"></input>*

*</div>*

*<div className={styles.itemField}>*

*<div className={styles.fieldLabel}>Software Name</div>*

*<input type="text" id="name"></input>*

*</div>*

*<div className={styles.itemField}>*

*<div className={styles.fieldLabel}>All Items:</div>*

*<div id="allItems"></div>*

*</div>*

*<div className={styles.buttonSection}>*

*<div className={styles.button}>*

*<span className={styles.label} onClick={this.createItem}>*

*Create*

*</span>*

*</div>*

*<div className={styles.button}>*

*<span className={styles.label} onClick={this.getItemById}>*

*Read*

*</span>*

*</div>*

*<div className={styles.button}>*

*<span className={styles.label} onClick={this.getAllItems}>*

*Read All*

*</span>*

*</div>*

*<div className={styles.button}>*

*<span className={styles.label} onClick={this.updateItem}>*

*Update*

*</span>*

*</div>*

*<div className={styles.button}>*

*<span className={styles.label} onClick={this.deleteItem}>*

*Delete*

*</span>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

**Step 11:**Import PNP modules in the **SpfxCrudReactPnp.tsx**file as shown below

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

*import "@pnp/sp/webs";*

*import "@pnp/sp/lists";*

*import "@pnp/sp/items";*

**Step 12:**Add CRUD functions as shown below in the same tsx file:

*//Create Item*

*private createItem = async () => {*

*try {*

*const addItem = await sp.web.lists.getByTitle("Test").items.add({*

*Title: document.getElementById("title")["value"],*

*SoftwareName: document.getElementById("name")["value"]*

*});*

*console.log(addItem);*

*alert(`Item created successfully with ID: ${addItem.data.ID}`);*

*} catch (e) {*

*console.error(e);*

*}*

*}*

*//Get Item by ID*

*private getItemById = async () => {*

*try {*

*const id: number = document.getElementById("itemId")["value"];*

*if (id > 0) {*

*const item: any = await sp.web.lists*

*.getByTitle("Test")*

*.items.getById(id)*

*.get();*

*document.getElementById("title")["value"] = item.Title;*

*document.getElementById("name")["value"] = item.SoftwareName;*

*} else {*

*alert(`Please enter a valid item id.`);*

*}*

*} catch (e) {*

*console.error(e);*

*}*

*}*

*//Get all items*

*private getAllItems = async () => {*

*try {*

*const items: any[] = await sp.web.lists.getByTitle("Test").items.get();*

*console.log(items);*

*if (items.length > 0) {*

*var html = `<table><tr><th>ID</th><th>Title</th><th>SoftwareName</th></tr>`;*

*items.map((item, index) => {*

*html += `<tr><td>${item.ID}</td><td>${item.Title}</td><td>${item.SoftwareName}</td></li>`;*

*});*

*html += `</table>`;*

*document.getElementById("allItems").innerHTML = html;*

*} else {*

*alert(`List is empty.`);*

*}*

*} catch (e) {*

*console.error(e);*

*}*

*}*

*//Update Item*

*private updateItem = async () => {*

*try {*

*const id: number = document.getElementById("itemId")["value"];*

*if (id > 0) {*

*const itemUpdate = await sp.web.lists*

*.getByTitle("Test")*

*.items.getById(id)*

*.update({*

*Title: document.getElementById("title")["value"],*

*SoftwareName: document.getElementById("name")["value"],*

*});*

*console.log(itemUpdate);*

*alert(`Item with ID: ${id} updated successfully!`);*

*} else {*

*alert(`Please enter a valid item id.`);*

*}*

*} catch (e) {*

*console.error(e);*

*}*

*}*

*//Delete Item*

*private deleteItem = async () => {*

*try {*

*const id: number = parseInt(document.getElementById("itemId")["value"]);*

*if (id > 0) {*

*let deleteItem = await sp.web.lists*

*.getByTitle("Test")*

*.items.getById(id)*

*.delete();*

*console.log(deleteItem);*

*alert(`Item ID: ${id} deleted successfully!`);*

*} else {*

*alert(`Please enter a valid item id.`);*

*}*

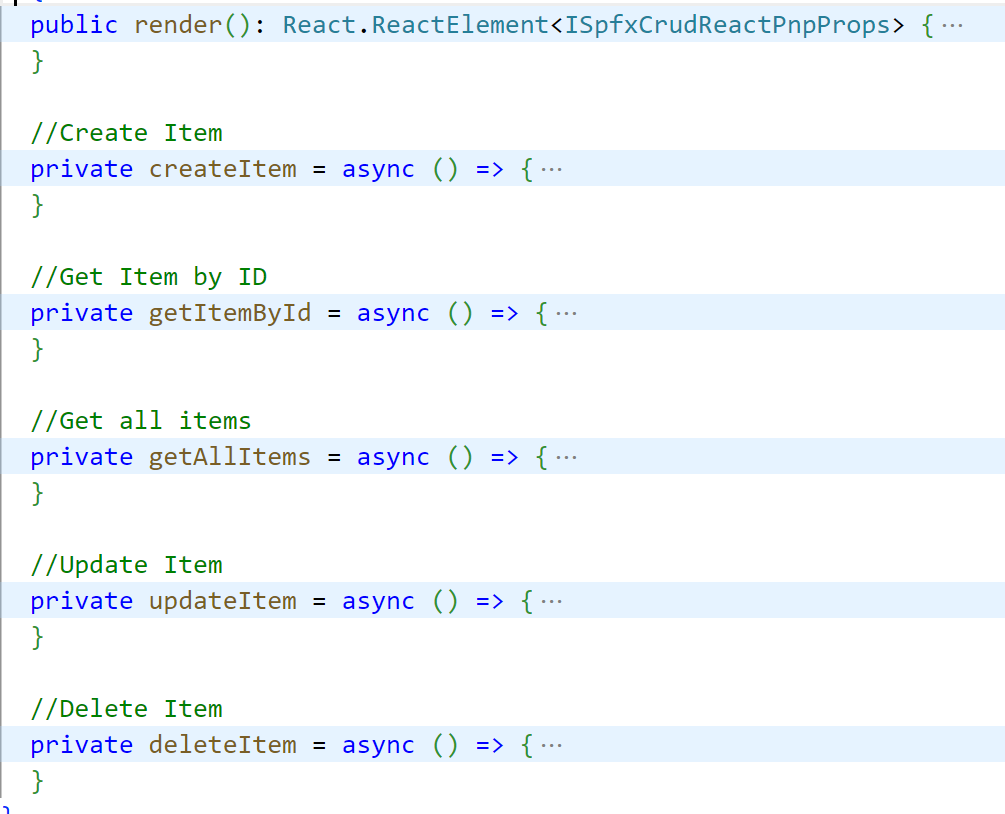
*} catch (e) {*

*console.error(e);*

*}*

*}*

**Step 13:**After adding all the above functions, your **SpfxCrudReactPnp.tsx**file will look like below:



**Step 14:**Update your **SpfxCrudReactPnp.module.scss**file with below CSS

*@import "~office-ui-fabric-react/dist/sass/References.scss";*

*.spfxCrudReactPnp {*

*.container {*

*max-width: 700px;*

*margin: 0px auto;*

*box-shadow: 0 2px 4px 0 rgba(0, 0, 0, 0.2), 0 25px 50px 0 rgba(0, 0, 0, 0.1);*

*}*

*.row {*

*@include ms-Grid-row;*

*@include ms-fontColor-white;*

*background-color: $ms-color-themeDark;*

*padding: 20px;*

*}*

*.column {*

*@include ms-Grid-col;*

*@include ms-lg10;*

*@include ms-xl8;*

*@include ms-xlPush2;*

*@include ms-lgPush1;*

*}*

*.title {*

*@include ms-font-xl;*

*@include ms-fontColor-white;*

*}*

*.subTitle {*

*@include ms-font-l;*

*@include ms-fontColor-white;*

*}*

*.description {*

*@include ms-font-l;*

*@include ms-fontColor-white;*

*}*

*.button {*

*// Our button*

*text-decoration: none;*

*height: 32px;*

*// Primary Button*

*min-width: 80px;*

*background-color: $ms-color-themePrimary;*

*border-color: $ms-color-themePrimary;*

*color: $ms-color-white;*

*// Basic Button*

*outline: transparent;*

*position: relative;*

*font-family: "Segoe UI WestEuropean", "Segoe UI", -apple-system,*

*BlinkMacSystemFont, Roboto, "Helvetica Neue", sans-serif;*

*-webkit-font-smoothing: antialiased;*

*font-size: $ms-font-size-m;*

*font-weight: $ms-font-weight-regular;*

*border-width: 0;*

*text-align: center;*

*cursor: pointer;*

*display: inline-block;*

*margin: 10px;*

*.label {*

*font-weight: $ms-font-weight-semibold;*

*font-size: $ms-font-size-m;*

*height: 32px;*

*line-height: 32px;*

*margin: 0 4px;*

*vertical-align: top;*

*display: inline-block;*

*}*

*}*

*.itemField {*

*display: flex;*

*padding: 5px;*

*.fieldLabel {*

*min-width: 100px;*

*}*

*}*

*.buttonSection {*

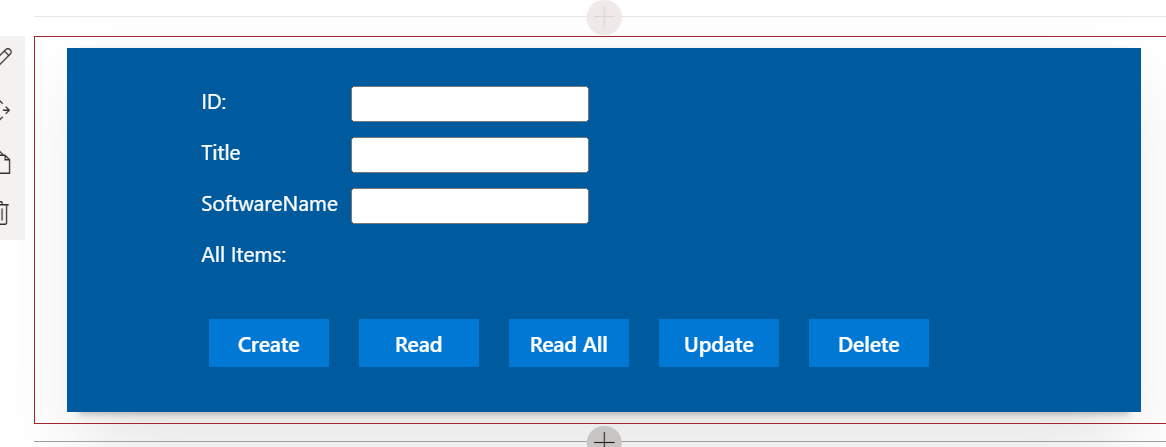
*padding-top: 20px;*

*display: flex;*

*}*

*}*

**Step 15:**Run the gulp serve. Open the SharePoint workbench. Our UI will look as below:



Using this form, we can perform the CRUD operation on the SharePoint list in SPFx using PNP.

## Note: Issue with PnP package:

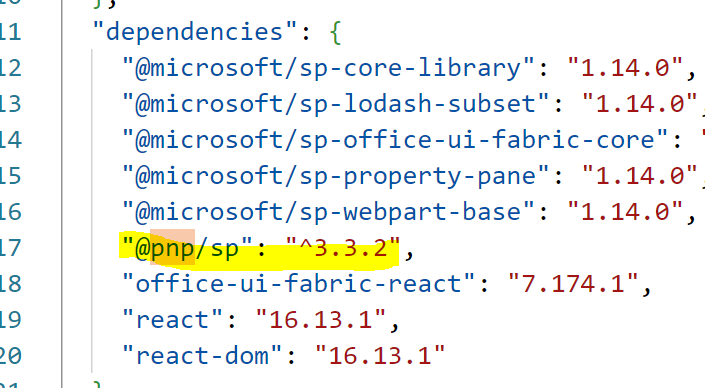
If you get error:

Module '"@pnp/sp"' has no exported member 'sp'

You need to upgrade as mentioned in the link:

<https://pnp.github.io/pnpjs/transition-guide/>

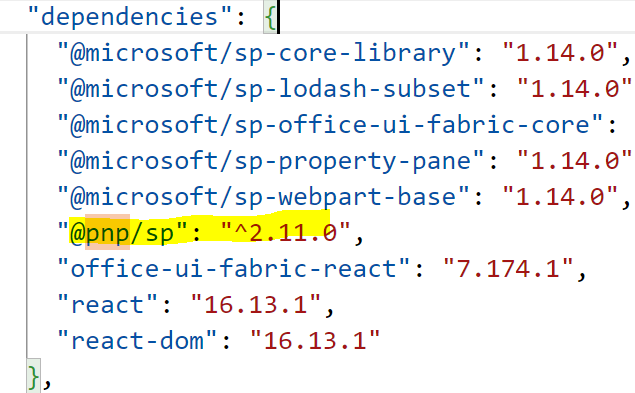
In package.json we have latest version of pnp



* This is the PnPjs version 3
* In PnPjs version 3 the**sp** and **graph**objects have been deprecated and will need to be replaced by the new objects like **spfi**and **graphfi**– In next article I will demonstrate spfi object
* Still there are few options if we dont want to go with latest version and continue with our older version of PnPjs and one of them is
  + Uninstall the recent / latest version of PnPjs
  + Install Version 2 of PnPjs as

*npm uninstall @pnp/sp*

*npm i @pnp/sp@2.11.0*



<https://knowledge-junction.com/2022/03/24/spfx-resolving-error-module-pnp-sp-has-no-exported-member-sp-ts2305/>