\* To use pnp install its pkg with following command

npm install sp-pnp-js –save

\*write gulp serve

\*add import in .ts file as given below

import { sp, List, Item, ItemAddResult, ItemUpdateResult } from 'sp-pnp-js';

write more pkgs name in brackets,

\*write your crud code in your export class e.g. mine is

export default class PropertiesWebpartWebPart extends BaseClientSideWebPart <IPropertiesWebpartWebPartProps>

CRUD CODE:

  public createItem():void

  {

    sp.web.lists.getByTitle("Students").items.add({

      Title: "hello"

    }).then((iar: ItemAddResult) => {

      console.log(iar);

    })

  }

  private readItem():void{

      // get a specific item by id

      sp.web.lists.getByTitle("Students").items.getById(1).get().then((item: any) => {

        console.log(item);

      })

    }

    private updateItem():void{

        let list = sp.web.lists.getByTitle("Students");

        list.items.getById(4).update({

            Title: "My New Title - SPFx0000"

        }).then(i => {

            console.log("ok",i);

        });

      }

private deleteItem():void{

    let list = sp.web.lists.getByTitle("Students");

    list.items.getById(1).delete().then(\_ => {console.log('List Item Deleted')});

  }

\* Then create it html in render() method

this.domElement.innerHTML = `<button  id="createButton">

    <span class="${styles.label}">Create item</span>

  </button> <br><br><br>

  <button id="readItem">

    <span>Read item</span>

  </button>

  <br><br><br>

  <button id="updateItem">

    <span>Update item</span>

  </button>

<br><br><br>

<button id="deleteButton">

   <span>Delete item</span>

   </button>

   <br><br><br>

  <a href="https://aka.ms/spfx" >

    <span >Learn more</span>

  </a>`;

\*In a separate function, I have written Event handlers for each button. In query selector id’s of buttons are written i.e. #createButton, #readItem, #updateItem, #deleteButton

private setButtonsEventHandlers(): void {

    console.log("yup");

    const webPart: PropertiesWebpartWebPart = this;

    this.domElement.querySelector('#createButton').addEventListener('click', this.createItem);

    console.log("create item ..............");

    this.domElement.querySelector('#readItem').addEventListener('click', this.readItem);

    console.log("reading item ..............");

    this.domElement.querySelector('#updateItem').addEventListener('click', webPart.updateItem);

    console.log("updating item ..............");

    this.domElement.querySelector('#deleteButton').addEventListener('click', webPart.deleteItem);

    console.log("deleteing item ..............");

  }

Functions are called in addEventListner with reference of webpart object. i.e.this.creatItem.

addEventListener('click', this.createItem)

\*Call Event handler function in render() function

 this.setButtonsEventHandlers();