**Lab 8 – Create a Webpart with TypeScript with JSON as Data source**

1. To create a new web part project

* Create a new project directory in your favourite location.
* Open command prompt
  + Run as administrator

***md TypeScriptwithJSONDB***

1. Go to the project directory

*cd* ***TypeScriptwithJSONDB***

1. Create a new ***TypeScriptwithJSONDB*** by running the Yeoman SharePoint Generator.

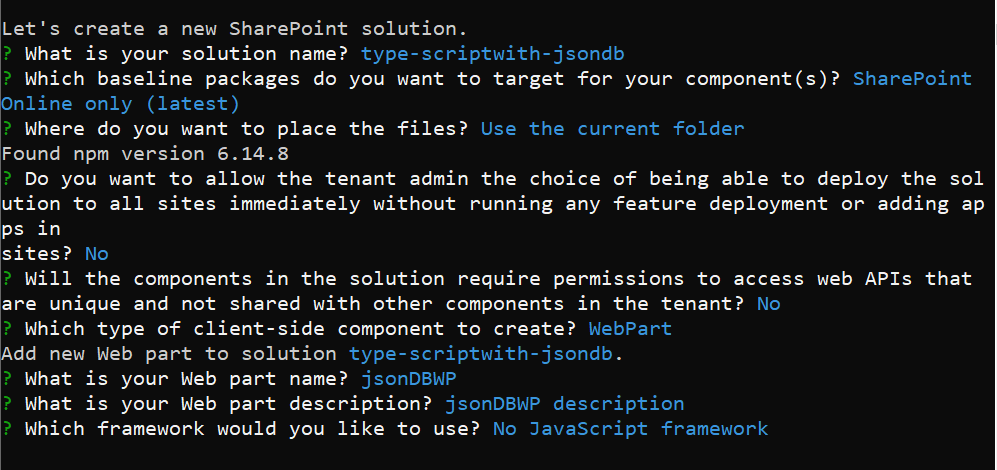
yo @microsoft/sharepoint

**When prompted**:

* Accept the default **type-scriptwith-jsondb** as your solution name, and then select Enter.
* Select SharePoint Online only (latest) and select Enter.
* Select Use the current folder for where to place the files.
* Select N to not allow the solution to be deployed to all sites immediately.
* Select N on the question if solution contains unique permissions.
* Select WebPart as the client-side component type to be created.

**The next set of prompts ask for specific information about your web part:**

* Enter **jsonDBWP** as your web part name, and then select Enter.
* Accept the default jsonDBWP description as your web part description, and then select Enter.
* Accept the default No JavaScript web framework as the framework you would like to use, and then select Enter



Open the solution using visual studio code using **code .**

* Create a folder **Planets** under Webparts
* Add new file GetPlanets.ts
* Add below code

export class getPlannets {

  public listAllPlannets():string[]

  {

    const planets: any = require("./Planets.json");

    const myObjStr = JSON.stringify(planets);

    let plannetNames:string[] = [];

    JSON.parse(myObjStr, (key, value:string) => {

      if (key === 'name') {

        plannetNames.push(value);

      }

    });

    return plannetNames;

  }

  public getPlannetDetails(plannetname:string)

  {

    //alert(plannetname);

    //document.getElementById("details").innerHTML = plannetname;

    const planets: any = require("./Planets.json");

    const selectedPlanet: any = planets.filter((planet) => planet.name === plannetname)[0];

    //const myObjStr = JSON.parse(planets);

    console.log(selectedPlanet);

    let displayPlannetdetails:string = `<table border='1'>

    <tr><td colspan=2><a href=${selectedPlanet.wikiLink} target=\_blank><img style="height:500px" src=${selectedPlanet.imageLink}></a></td></tr>

    <tr><td width="40%">ID</td><td width="60%">${selectedPlanet.id}</td></tr>

    <tr><td>Name</td><td>${selectedPlanet.name}</td></tr>

    <tr><td>Summary</td><td>${selectedPlanet.summary}</td></tr>

    <tr><td>Solar Orbit Years</td><td>${selectedPlanet.solarOrbitYears}</td></tr>

    <tr><td>Solar Orbit Avg Distance Km</td><td>${selectedPlanet.solarOrbitAvgDistanceKm}</td></tr>

    <tr><td>Num Satellites</td><td>${selectedPlanet.numSatellites}</td></tr>

    `;

    displayPlannetdetails += `</table>`;

    document.getElementById("details").innerHTML = displayPlannetdetails;

  }

}

* Add **Planets.json** file under the same folder (download from github)
* Then go to webpart file **JsonDbwpWebPart.ts**
* Import the module

import \* as plannets from './../Planets/GetPlanets';

* Then replace the render method with below code

public render(): void {

  let getplanet = new plannets.getPlannets();

  let plannetNames:string[] = getplanet.listAllPlannets();

  let listplannetnames:string = "<ul>";

  for (let i = 0; i < plannetNames.length; i++) {

    listplannetnames += '<li><button type=button id="btn" class="btn" value=' + plannetNames[i] +'>' + plannetNames[i] + '</button></li>';

  }

  listplannetnames += "</ul>";

  this.domElement.innerHTML = `

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

  <div class="${ styles.container }">

    <div class="${ styles.row }">

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

        <span class="${ styles.title }">${listplannetnames}</span>

        <div id=details></div>

        </div>

        </div>

        </div>

        </div>`;

        this.\_setButtonEventHandlers();

}

Then Add below private method next to render method

private \_setButtonEventHandlers(): void {

  //const button:HTMLButtonElement = document.querySelector(".btn");

  //let getplanet = new plannets.getPlannets();

  //let plannetname : string = button.value;

  //button.addEventListener("click", () => {getplanet.getPlannetDetails(plannetname);});

  let getplanet = new plannets.getPlannets();

  document.querySelectorAll('.btn').forEach(item => {

    let plannetname : string = item.innerHTML;

    item.addEventListener('click', event => {

      getplanet.getPlannetDetails(plannetname);

    });

  });

}

Output

