**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.

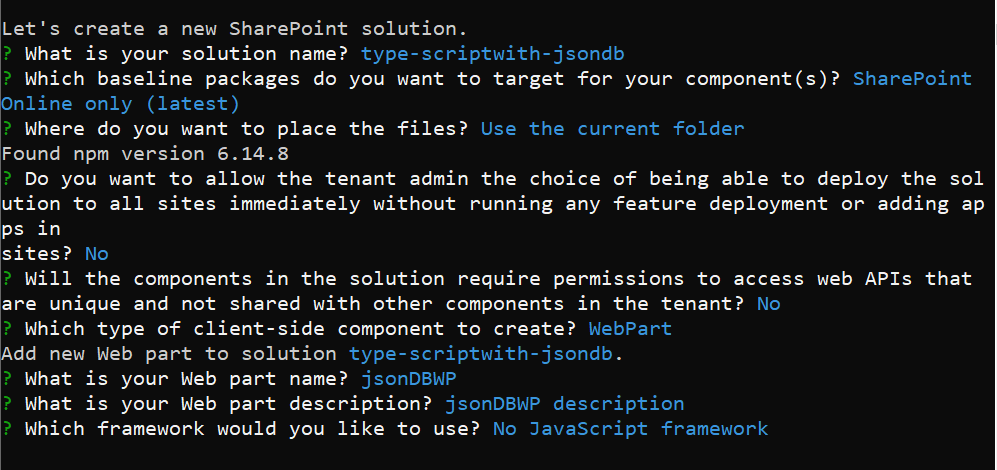
yo @microsoft/sharepoint

**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.

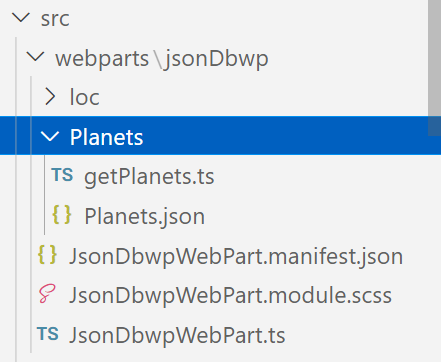
**The next set of prompts ask for specific information about your web part:**

* 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 **Planets.json** file under the same folder (download from github)
* Add new file **GetPlanets.ts**
* Add below code

export class getPlanets {

    public listAllPlanets(): string[] {

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

        const myObjStr = JSON.stringify(planets);

        let planetsNames: string[] = [];

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

            if (key === 'name') {

                planetsNames.push(value);

            }

        });

        return planetsNames;

    }

    public getPlanetDetails(planetname: string) {

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

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

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

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

    </table>`;

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

    }

}

* Then go to webpart file **JsonDbwpWebPart.ts**
* Import the module

import \* as planets from './Planets/getPlanets';

* Then replace the render method with below code

public render(): void {

    let getPlanet = new planets.getPlanets();

    let planetsNames:string[] = getPlanet.listAllPlanets();

    let listplanetname:string = "<div>";

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

      listplanetname += '<span><button type=button id="btn" class="btn" value=' + planetsNames[i] + '>' + planetsNames[i] + '</button></span><br>' ;

      }

      listplanetname += '</div>';

    this.domElement.innerHTML = `

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

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

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

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

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

              <div id="details"/>

            </div>

          </div>

        </div>

      </div>`;

      this.\_setButtonEventHandlers();

  }

Then Add below private method next to render method

private \_setButtonEventHandlers():void{

    let getPlanet = new planets.getPlanets();

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

      let planetname:string = item.innerHTML;

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

        getPlanet.getPlanetDetails(planetname);

      });

    });

  }

Output

