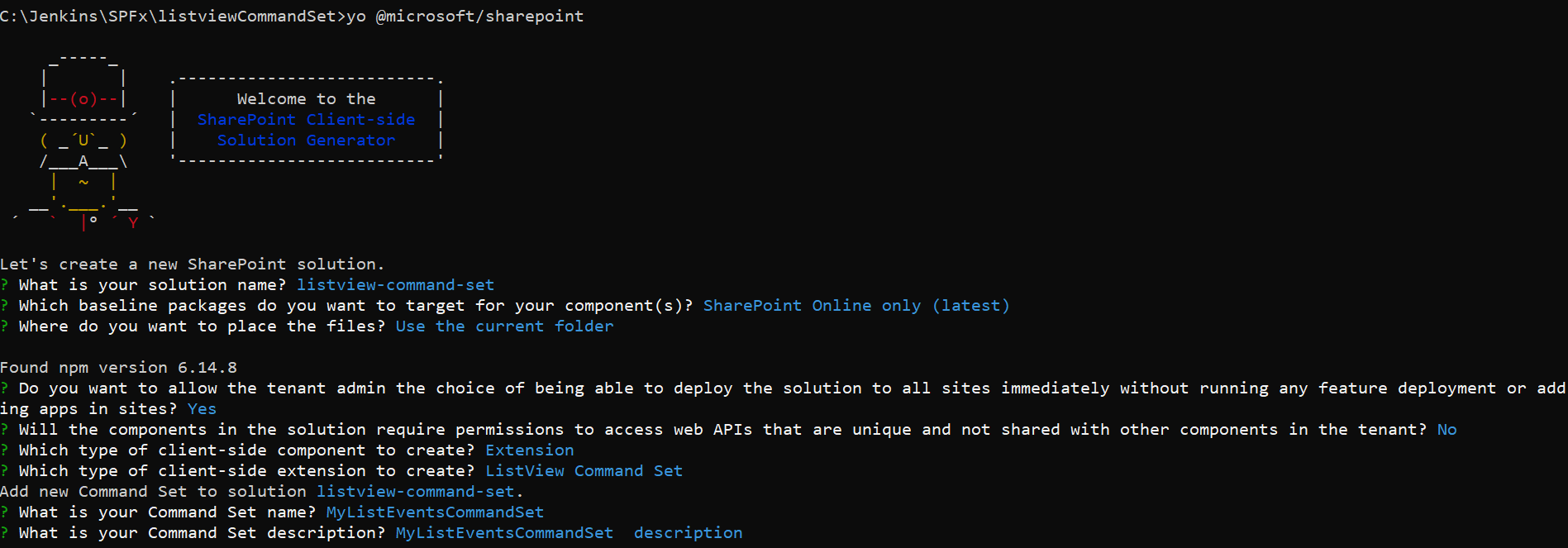
**Commend Set Customizer**

* Open Command prompt
* Navigate to your favourite folder
* Ex: C:\Jenkins\SPFx\Nodejs\SPFx\extension
* Create new folder for app customizer
* Ex: listViewCommandSetEx1
* Create spfx solution >yo @microsoft/sharepoint



Go to code .

Open the serve.json file under config and add any list view URL ex:

https://adminspfxibm.sharepoint.com/Lists/MyProducts/AllItems.aspx

Then open MyListEventsCommandSetCommandSet.ts file

Replace onListViewUpdated method

public onListViewUpdated(event: IListViewCommandSetListViewUpdatedParameters): void {

    const \_GetItem: Command = this.tryGetCommand('GetItemButton');

    const \_GetItems: Command = this.tryGetCommand('MultipleItemButtom');

    const \_alertbutton: Command = this.tryGetCommand('AlertMessageButton');

    const \_deletebtn: Command = this.tryGetCommand('DeleteButton');

    if (\_GetItem) {

      \_GetItem.visible = event.selectedRows.length === 1;

    }

    if (\_GetItems) {

      \_GetItems.visible = event.selectedRows.length > 0;

    }

    if (\_alertbutton) {

      \_alertbutton.visible = event.selectedRows.length === 0;

    }

    if (\_deletebtn) {

      \_deletebtn.visible = event.selectedRows.length > 0;

    }

  }

Then update/replace Execute method

 public onExecute(event: IListViewCommandSetExecuteEventParameters): void {

    switch (event.itemId) {

      case 'GetItemButton':

        Dialog.alert(`Selected Item : ${event.selectedRows[0].getValueByName("Title")}`);

        break;

      case 'MultipleItemButtom':

        this.multipleitemevent(event);

        break;

      case 'AlertMessageButton':

        Dialog.alert(`No Items Selected Button`);

        break;

      case 'DeleteButton':

        //Dialog.alert(event.selectedRows[0].getValueByName('ID'));

        this.deletethisitem(event.selectedRows[0].getValueByName('ID'));

        break;

      default:

        throw new Error('Unknown command');

    }

  }

Add below functions

private deletethisitem(id: string) {

    return new Promise<{}>((resolve, reject) => {

      const endpoint: string = `${this.context.pageContext.web.absoluteUrl}/\_api/lists/getbytitle('Demo list')/items(${id})`;

      this.context.spHttpClient.post(endpoint, SPHttpClient.configurations.v1, {

        headers: {

          'Accept': 'application/json;odata=nometadata',

          'Content-type': 'application/json;odata=verbose',

          'odata-version': '',

          'IF-MATCH': '\*',

          'X-HTTP-Method': 'DELETE'

        }

      })

        .then((response: SPHttpClientResponse): void => {

          alert(`Item with ID: ${id} successfully Deleted`);

        }, (error: any): void => {

          alert(`${error}`);

        });

    });

  }

  private multipleitemevent(event: IListViewCommandSetExecuteEventParameters) {

    let title = "";

    for (var i = 0; i < event.selectedRows.length; i++) {

      title += " " + event.selectedRows[i].getValueByName("Title") + " ";

    }

    Dialog.alert(`${title}`);

  }

Then update the **MyListEventsCommandSetCommandSet.manifest.json** file items section

"items": {

        "GetItemButton": {

            "title": { "default": "Get Item" },

            "iconImageUrl": "icons/request.png",

            "type": "command"

        },

        "AlertMessageButton": {

            "title": { "default": "No Items Selected" },

            "iconImageUrl": "icons/cancel.png",

            "type": "command"

        },

        "MultipleItemsButton": {

            "title": { "default": "Multiple Items" },

            "iconImageUrl": "icons/cancel.png",

            "type": "command"

        },

        "DeleteButton": {

            "title": { "default": "My Delete Button" },

            "iconImageUrl": "icons/cancel.png",

            "type": "command"

        }

    }

Gulp serve and test it