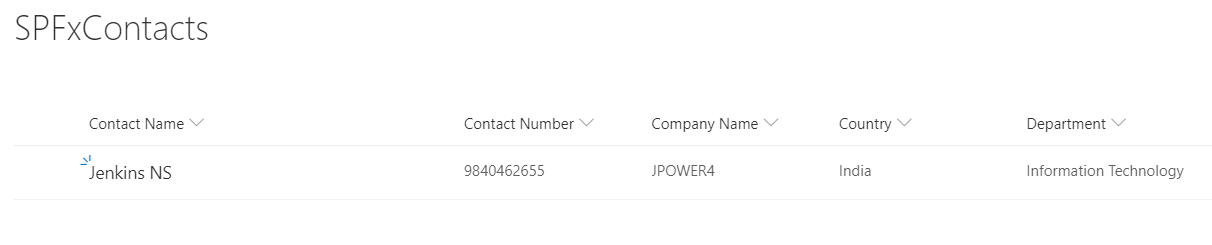
Consume Data from SharePoint using Graph API

1. Create a SharePoint list “SPFxContacts” with below columns

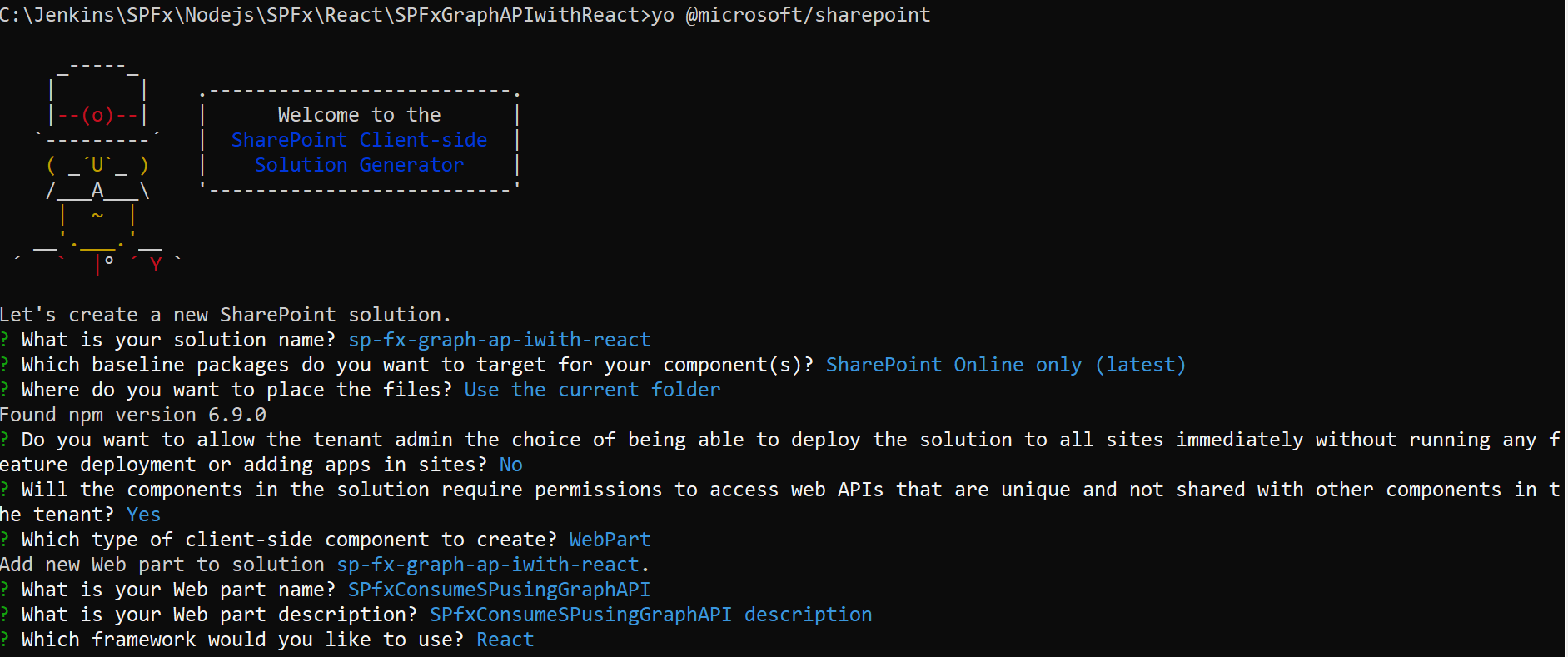
<https://jpower4mvp.sharepoint.com/sites/SPFxContacts/AllItems.aspx>

* 1. Title – already available – rename to Contact Name
  2. Contac Number – Single line Text
  3. Company Name – Single line Text
  4. Country – Single line Text



1. Create SPfx webpart

* Navigate to your favourite folder
* Create new folder “SPFxGraphAPIwithReact”
* Navigate to new folder “SPFxGraphAPIwithReact”
  + Create a new web part by running the Yeoman SharePoint Generator
  + yo @microsoft/sharepoint



* Type code . and Open the webpart in Visual studio code
* Go to ./src/webparts/ sPfxConsumeSPusingGraphApi/components
* Create new file “IListItem.ts” and copy and paste below code

export interface IListItem {

  Title: string;

  ContactNumber: string;

  CompanyName: string;

  Country:string;

  }

* Then create new file “ISPfxConsumeSPusingGraphApiState.ts” and copy and paster below code.

import { IListItem } from './IListItem';

export interface ISPfxConsumeSPusingGraphApiState {

  lists: Array<IListItem>;

}

* Then open ISPfxConsumeSPusingGraphApiProps.ts file and add below import statement

import { WebPartContext } from '@microsoft/sp-webpart-base';

* Then include one more variable context

context: WebPartContext; in the inteface

* Then open SPfxConsumeSPusingGraphApi.tsx file
* Add below import statements

import { ISPfxConsumeSPusingGraphApiState } from './ISPfxConsumeSPusingGraphApiState';

import { MSGraphClient } from "@microsoft/sp-http";

import { IListItem } from './IListItem';

import {

  autobind,

  PrimaryButton,

  TextField,

  Label,

  DetailsList,

  DetailsListLayoutMode,

  CheckboxVisibility,

  SelectionMode

} from 'office-ui-fabric-react';

* Then add list item columns to display using detailslist

// Configure the columns for the DetailsList component

let \_listItemColumns = [

  {

    key: 'ContactPerson',

    name: 'Contact Person',

    fieldName: 'Title',

    minWidth: 50,

    maxWidth: 200,

    isResizable: true

  },

  {

    key: 'ContactNumber',

    name: 'Contact Number',

    fieldName: 'ContactNumber',

    minWidth: 50,

    maxWidth: 200,

    isResizable: true

  },

  {

    key: 'CompanyName',

    name: 'Company Name',

    fieldName: 'CompanyName',

    minWidth: 50,

    maxWidth: 200,

    isResizable: true

  },

  {

    key: 'Country',

    name: 'Country',

    fieldName: 'Country',

    minWidth: 50,

    maxWidth: 200,

    isResizable: true

   }

];

* Add the parameter for SPfxConsumeSPusingGraphApi class like below

export default class SPfxConsumeSPusingGraphApi extends React.Component<ISPfxConsumeSPusingGraphApiProps,ISPfxConsumeSPusingGraphApiState, {}> {

* Replace the render method <div className={styles.column}>…</div> with below code

<div className={ styles.column }>

              <span className={ styles.title }>List Contact List Items</span>

          {

            (this.state.lists != null && this.state.lists.length > 0) ?

              <p className={ styles.form }>

              <DetailsList

                  items={ this.state.lists }

                  columns={ \_listItemColumns }

                  setKey='set'

                  checkboxVisibility={ CheckboxVisibility.hidden }

                  selectionMode={ SelectionMode.none }

                  layoutMode={ DetailsListLayoutMode.fixedColumns }

                  compact={ true }

              />

            </p>

            : null

          }

</div>

* Add the constructor inside the class

constructor(props: ISPfxConsumeSPusingGraphApiProps, state: ISPfxConsumeSPusingGraphApiState) {

    super(props);

    // Initialize the state of the component

    this.state = {

      lists: []

    };

  }

* Then add componentDidMount() method to access SharePoint using Graph API

public componentDidMount(){

    // Log the current operation

    console.log("Using \_searchWithGraph() method");

    this.props.context.msGraphClientFactory

      .getClient()

      .then((client: MSGraphClient): void => {

        client

          .api("sites('root')/lists(SPFxContacts)/items?expand=fields")

          .version("v1.0")

          .get((err, res) => {

            if (err) {

              console.error(err);

              return;

            }

            // Prepare the output array

            var lists: Array<IListItem> = new Array<IListItem>();

            // Map the JSON response to the output array

            res.value.map((item: any) => {

              lists.push({

                Title: item.fields.Title,

                ContactNumber: item.fields.ContactNumber,

                CompanyName: item.fields.CompanyName,

                Country: item.fields.Country

              });

            });

            // Update the component state accordingly to the result

            this.setState(

              {

                lists: lists,

              }

            );

          });

      });

  }

* Then go to SPfxConsumeSPusingGraphApiWebPart.ts file
* Add the context next to the description in render method

description: this.properties.description,

**context: this.context**

Then open SPfxConsumeSPusingGraphApi.module.scss and add below class next to .title

.form {

    @include ms-font-l;

    @include ms-fontColor-white;

}

Then open package-solution.json file under config

Paste below code next to “isDomainIsolated”:true,

"skipFeatureDeployment": true,

        "webApiPermissionRequests": [{

                "resource": "Microsoft Graph",

                "scope": "User.ReadBasic.All"

            },

            {

                "resource": "Microsoft Graph",

                "scope": "Sites.Read.All"

            },

            {

                "resource": "Microsoft Graph",

                "scope": "Sites.ReadWrite.All"

            }

        ]

**Test the webpart**

1. Open commend prompt **gulp serve**
2. **gulp build**
3. **gulp bundle --ship**
4. **gulp package-solution --ship**
5. Deploy the webpart in app catalog i.e upload **sp-fx-graph-ap-iwith-react.sppkg** fine from .\SPFx\React\SPFxGraphAPIwithReact\sharepoint\solution\
6. The install the webpart in the spfxtraining site and add the webpart
7. Go to SharePoint admin center <https://jpower4mvp-admin.sharepoint.com/_layouts/15/online/AdminHome.aspx#/home>
8. Navigate Advanced 🡪 API access 🡪 approve the graphapi