**How to build PCF Dataset Control with React**

**Creating a new project**

Once your environment is ready, create a new PCF dataset control project.

1. Open **visual studio code**, and from the terminal navigate to the folder where you want to create the PCF control project
2. **Run the following command** from your terminal to create a new PCF control project:

pac pcf init --namespace SampleNamespace --name ReactDatasetControl --template dataset --framework react --run-npm-install

* --namespace specifies the namespace for your control.
* --name specifies the name of your control.
* --template specifies the type of control (e.g., field or dataset)
* --framework (optional) specifies the framework for the control.
* --run-npm-install installs the required node modules for the control.

3. Running the above pac pcf init command sets up a basic PCF control with all the required files and dependencies, making it ready for you to customize and deploy to PowerApps.

4. In this tutorial we’ll be using the XLSX library to work with excel files. Install them in your project using,

npm install xlsx  
npm install --save-dev ajv

npm i @fluentui/react

**Overview**

Now that you’re set up, let’s get an overview of PCF Dataset Control!

**Inspecting the starter code**

In the explorer you’ll see three main sections:

A screenshot of a computer program

AI-generated content may be incorrect.

1. *node\_modules*contains all the node packages required for the project.
2. *ReactDatasetControl* project folder contains list of files like HelloWorld.tsx ,ControlManifest.Input.xml ,index.ts
3. eslintrc.json ,package.json are the configuration files of the project.

Let’s have a look at some of the key files.

ControlManifest.Input.xml

It’s where you define the configuration and properties of your PCF control. It includes information such as the control’s name, version, description, and the data types it will accept.

index.ts

It serves as the entry point for your PCF control’s business logic. It is where you define the behavior and interactions of your control.

A computer screen shot of text

AI-generated content may be incorrect.

**Lifecycle Methods**:

* **init**: This method is called when the control is initialized. It is used to set up the control, including event handlers, and to render the initial UI.
* **updateView**: This method is called whenever the control needs to be updated, for example, when the control's data or properties change. It is used to re-render the UI with the latest data.
* **getOutputs**: This method returns the current value of the control's outputs, which can be used by other components or stored in the database.
* **destroy**: This method is called when the control is removed from the DOM. It is used to clean up resources, such as event handlers, to prevent memory leaks.

In our control, the updateView method is called when the page loads. This method renders the HelloWorld component from the HelloWorld.tsx file.

**Building the component**

Let’s start by creating a App.tsx file inside the *ReactDatasetControl*project folder. This will consist the logic of our PCF control.

/\* App.tsx \*/  
  
import \* as React from 'react';  
import { IInputs } from "./generated/ManifestTypes";  
import { DetailsList } from '@fluentui/react';  
  
export function PCFControl({sampleDataSet} : IInputs) {  
  
 const records = [  
 {  
 "First Name": "Saturo",  
 "Last Name": "Gojo",  
 "Domain": "Infinity Void"  
 },  
 {  
 "First Name": "Sukuna",  
 "Last Name": "Ryomen",  
 "Domain": "Malevolent Shrine"  
 }  
 ];  
  
 return <DetailsList items={records}/>  
}

* Lines 1–3 brings all the necessary imports for the control.
* Next line defines a function called PCFControl . The export JavaScript keyword makes this function accessible outside of this file.
* In the PCFControl function a sampleDataSet property is passed that will consist the grid data/records to which our control will be attached.
* The function returns a DetailsList control with the records data passed to it’s items property.
* DetailsList is a [Fluent UI](https://developer.microsoft.com/en-us/fluentui#/) react control created by Microsoft that is used to view data in List format.

index.ts

Open the file labeled as index.ts , import our PCFControl at the top of the file and let’s modify the updateView function to return our custom control.

import { PCFControl } from "./App"  
  
// ...  
  
public updateView(context: ComponentFramework.Context<IInputs>): React.ReactElement   
{  
 const props: IInputs = { sampleDataSet: context.parameters.sampleDataSet };  
 return React.createElement(PCFControl, props);  
}  
  
// ...

context.parameters.sampleDataSet accesses the dataset property from the context parameter. It contains the grid data/records that our control will be bound to in later sections. Then we are returning a React element consisting of PCFControl component and passing dataset props as the properties to it.

Now that we have created a basic structure for our control, let’s build and execute our control to see how’s its looking. To view the control in your local browser execute the following commands in the terminal,

npm run build  
npm start watch

Now you can see your records in a tabular list format,

A screenshot of a computer

AI-generated content may be incorrect.