**Building a PCF Control that calls Context WebApi**

Power Apps Component Framework, or PCF Controls, have capabilities to call WebApi operations. we will look at how to build a PCF control that calls the Context WebApi.

In looking at the context.webAPI, we see we have the following methods:

* createRecord
* deleteRecord
* retrieveMultipleRecords
* retrieveRecord
* updateRecord

A screenshot of a computer program

AI-generated content may be incorrect.

Before we can use the WebAPI, we need to uncomment the feature-usage area in the ControlManifest.Input.xml file. It will look like below before you uncomment it:

A screenshot of a computer program

AI-generated content may be incorrect.

We will only include the <uses-feature name=”WebAPI” required=”true” /> line:

A screen shot of a computer program

AI-generated content may be incorrect.

Note without enabling this feature you may run into a “Feature is required to be specified in the <uses-feature> section in ControlManifest” error.

Let’s add some code. We will the following code to our index.ts, which will create a new account record:

*context.webAPI.createRecord("account", { name: "Sample Account" }).then(*

*function (success) {*

*console.log("Account created with ID: " + success.id);*

*},*

*function (error) {*

*console.log(error.message);*

*}*

*);*

A screen shot of a computer program

AI-generated content may be incorrect.

Now when we deploy this to our org and run this by pressing our button, we get “Account created with Id” in our console:

A screenshot of a computer

AI-generated content may be incorrect.



And our sample account is created:

A screenshot of a computer

AI-generated content may be incorrect.

Our webApi is working nicely! Let’s try the other methods.

To delete an account:

var Id = "860476ad-58ba-ee11-a569-000d3a3b9c87";

context.webAPI.deleteRecord("account", Id).then(() => {

alert("Account deleted");

}).catch((error) => {

alert(error.message);

});

This deletes the account:

A white background with black text

AI-generated content may be incorrect.

To retrieveMultiple accounts:

private outputdiv: HTMLDivElement;

context.webAPI.retrieveMultipleRecords("account", "?$select=name").then(

let outputdata ="";

        this.btn\_getAccount = document.createElement("button");

        this.btn\_getAccount.innerHTML = "Get All Accounts";

        this.btn\_getAccount.onclick =() => {

            context.webAPI.retrieveMultipleRecords("account","?$select=name").then(

                result => {

                    for(const i of result.entities){

                        outputdata = `${outputdata + i["name"]},`;

                    }

                    this.outputdiv = document.createElement("div");

                    this.outputdiv.innerHTML = outputdata;

                    container.appendChild(this.outputdiv);

                    return "";

                }).catch(e=>{console.log(e)})

        }

        container.appendChild(this.btn\_getAccount);

This returns:

A screenshot of a computer

AI-generated content may be incorrect.

To retrieve a single account:

var Id = "da59721f-02b2-ea11-a812-000d3a1b14a2";

context.webAPI.retrieveRecord("account", Id, "?$select=name").then(

function success(result) {

alert("Account Name: " + result["name"]);

},

function (error) {

alert(error.message);

}

);

This returns:

A close up of a logo

AI-generated content may be incorrect.

And to update an account:

var Id = "da59721f-02b2-ea11-a812-000d3a1b14a2";

context.webAPI.updateRecord("account", Id, { name: "Updated Account Name" }).then(

function success(result) {

console.log("Account name updated.");

},

function (error) {

console.log(error.message);

}

);

This produces an updated account record:



A screenshot of a computer

AI-generated content may be incorrect.

Those are the WebApi methods, you are now able to perform these in your PCF controls.