Data Validation

You can perform the PowerApps validation on submit, but as we mentioned before we can validate PowerApps when the user leaves the current field.

We will work on a SharePoint list with these columns:

* Employee Name -Requierd field
* Employee Id – must be a number
* Email Address -must match email form.
* Age- must be more than 18
* Phone- must match specific expression
* Required Field Validation In PowerApps

A screenshot of a computer

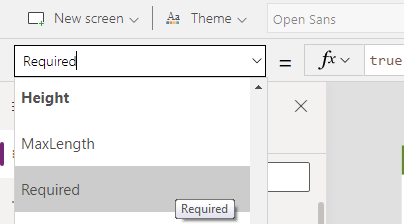
Description automatically generated

**For implementing PowerApps validation on submit while working with cards**

1. Select the card
2. From the advanced properities choose to unlock to change properties



1. On the Required property of the card change it to be true



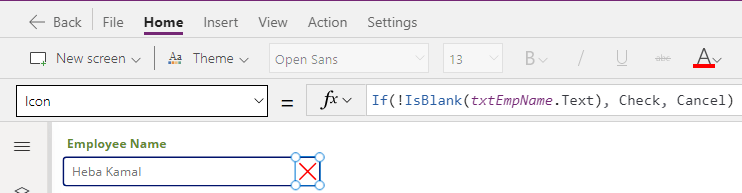
this is the powerapps required field validation on submit.

**For implementing PowerApps validation before submit**

In my example, I will insert an icon to show the user if there are any errors or not and I will write the error in the tooltip of the icon.

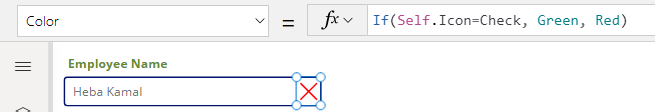
1. Insert an Icon in the employee name textbox
2. In the icon property write this formula

If(!IsBlank(txtEmpName.Text), Icon.Check, Icon.Cancel)



1. In the color property write this formula

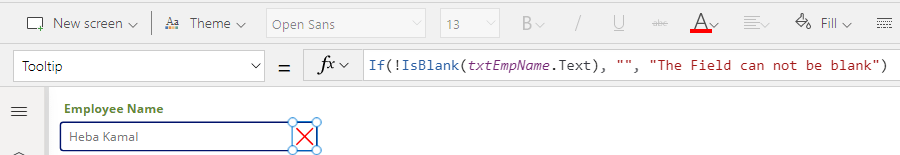
If(Self.Icon=Icon.Check, Green, Red)

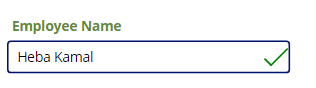


1. In the tooltip property write this formula

If(!IsBlank(txtEmpName.Text), "", "The Field can not be blank")

You can apply this step on a label insteed of the tooltip as we mentioned before

Now when the field is black the icon will be the X icon with the color red otherwise the icon will be the check icon with green color.



This is the difference between powerapps validation on submit and powerapps validation before submit.

**PowerApps Number Validation**

To validate a textbox to Accept only numbers:

1. Select the textbox
2. Make the Format property to be number
3. Also you can use the IsNumeric function to check if the value entered in the textbox is number or not.

IsNumeric( txtEmpID.text)

**PowerApps Validate Mobile Number**

1. Select the mobile textbox
2. Make the Format property to be number to allow only digits
3. Insert an icon and use the same method we used in the Employee Name

In the Icon property set the formula to allow only 10 digits

If(

IsMatch(

txtmobile.Text,

Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit

),

Icon.Check,

Icon.Cancel

)

and in the color property set the formula to be

If(Self.Icon=Icon.Check, Green, Red)

you can add an error message to the user either on a lable or on the tooltip property of the icon as shown bellow

If(

IsMatch(

txtmobile.Text,

Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit & Digit

),

"",

"the mobile must be 10 digit"

)

**Zip Code Field Validation**

The PowerApps Zip Code field Validation is the same as we have done in the previous example except for the number of the digits

**Validating Age Field**

In this PowerApps validation example, we’ll look at how to validate a number field in PowerApps to be > or < specific number.

let’s check if the age field is more than 21 years old.

1. Select the Age textbox
2. Make the Format property to be number to allow only digits
3. Insert an icon and use the same method we used in the Employee Name
4. In the Icon property set the formula to be

If(Value(txtAge.Text) >= 21, Icon.Check, Icon.Cancel)

and in the color property set the formula to be

If(Self.Icon=Icon.Check, Green, Red)

you can add an error message to the user either on a lable or on the tooltip property of the icon as shown below

If(Value(txtAge.Text) >= 21, "", "Age must be >=21")

**PowerApps Email Validation**

Another example of PowerApps validation before submit is validating Email address field. An email address must be in the format example@ example.com.

There is a predefined matching pattern already exists for email addresses that you can use

1. Select the Email Address textbox
2. Insert an Icon as we used before in the employee name .

In the Icon property set the formula to be

If(

IsMatch(txtEmail.Text, Match.Email),

Icon.Check,

Icon.Cancel

)

1. In the color property write this formula

If(Self.Icon=Icon.Check, Green, Red)

you can add the validation error message to the user either on a lable or on the tooltip property of the icon as shown bellow

If(

IsMatch(

txtEmail.Text,

Match.Email

),

"",

"the Email should be in the formate example@example.com"

)

**Data Validation With Pattern Matching**

1. What if we want to validate PowerApps textbox to be in a specific format, For example, we want to formate the phone number to be n the format ###-###-#### then the formula of the Icon property should be

If(

IsMatch(

txtPhoner.Text,

Match.Digit&Match.Digit&Match.Digit&"-"&

Match.Digit&Match.Digit&Match.Digit&"-"&

Match.Digit&Match.Digit&Match.Digit&Match.Digit

),

Icon.Check,

Icon.Cancel

)

1. and in the color property set the formula to be

If(Self.Icon=Icon.Check, Green, Red)

1. you can add the validation error message to the user either on a lable or on the tooltip property of the icon as shown bellow

If(

IsMatch(

txtphone.Text,

Match.Digit&Match.Digit&Match.Digit&"-"&

Match.Digit&Match.Digit&Match.Digit&"-"&

Match.Digit&Match.Digit&Match.Digit&Match.Digit

),

"",

"the Phone should be in the format ###-###-#### "

)

**PowerApps Validate Date**

1. Are you trying to validate the Date field in PowerApps? This PowerApps validation example shows how to validate a date field in PowerApps. The user must choose a weekday from Monday to Friday and must be a day in the future.
2. Insert an icon and in the icon property write the followin formula

If(

Weekday(joinDate.SelectedDate, StartOfWeek.Monday) <= 5

And joinDate.SelectedDate > Today(),

Icon.Check, Icon.Cancel

)

1. and in the color property set the formula to be

If(Self.Icon=Icon.Check, Green, Red)

1. you can add the validation error message to the user either on a lable or on the tooltip property of the icon as shown bellow

If(

Weekday(joinDate.SelectedDate, StartOfWeek.Monday) > 5,

"Choose a day from Monday to Friday",

joinDate.SelectedDate <= Today(),

"Must choose a date in the future",

"No date was selected"

)

**Validating Password In PowerApps**

1. Select the Password textbox
2. Insert an Icon as we used before in the employee name .
3. In the Icon property set the formula to be

Icon = If(

IsMatch(

txtPassword.Text,

"^(?=[^\d\_].\*?\d)\w(\w|[!@#$%]){7,20}

"

),

Check,

Icon.Lock

)

this validates a strong password which can contain eight, nine or 10 characters with the addition of at least one digit and at least one alphabetic character and no special charachters.

1. In the color property write this formula

If(Self.Icon=Icon.Check, Green, Red)

1. you can add the validation error message to the user either on a lable or on the tooltip property of the icon as shown bellow

If(

IsMatch(

txtEmail.Text,

Match.Email

),

"",

"the Email should be in the formate example@example.com"

)

Co

**Integrate Power Apps on Power BI Dashboard**

1. sample power bi dashboard with power app to add comments

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. design your power bi dashboard
2. add a power app visual from the visualization panel

A screenshot of a computer

Description automatically generated

1. Resize

A screenshot of a computer

Description automatically generated

1. Set a field for the power app

A screenshot of a computer

Description automatically generated

1. Then click ‘create new’

A screenshot of a computer

Description automatically generated

1. Then click ‘go to power app studio’ to start designing the app

A screenshot of a computer

Description automatically generated

1. In the power app studio, it will be shown like this (note of the gallery and the formula):

A screenshot of a computer

Description automatically generated

1. Delete the gallery then insert an edit form, connect the edit form to your data source

A screenshot of a computer

Description automatically generated

1. In the edit form’s **Item** property, set the ff:

LookUp(‘Lease Comments List’,Title=First(PowerBIIntegration.Data).LeaseID)

1. Toggle between edit mode and new form mode depending on the found rows. Set the ff formula in the **DefaultMode** property of the edit form:

If(CountRows(Filter(‘Lease Comments List’, Title=First(PowerBIIntegration.Data).LeaseID)) = 1,

FormMode.Edit,

FormMode.New)

1. In the Edit form, unlock and edit the LeaseID input text **Default** property to display the first record

First(PowerBIIntegration.Data).LeaseID

1. (\*optional), you can also set the **DefaultDate** property of the ActionDate input text to:

Now()

1. Add a button and use the ff code:

SubmitForm(Form1); Refresh(‘Lease Comments List’)

A screenshot of a computer

Description automatically generated

1. Lets add a cover to the form (rectangle then change color to white)

A screenshot of a computer

Description automatically generated

1. Change the rectangle color to white and add instruction text label (group the two controls afterwards)

A screenshot of a computer

Description automatically generated

1. Set the ff formula in the **Visible** property of the grouped rectangle and textLabel.

If(CountRows(PowerBIIntegration.Data))=1, false, true)

1. Save and Publish the app.
2. Check in your power bi dashboard, if the app is not automatically selected, choose it from the list of existing applications.

**Power Apps and Power Automate Integration**

1. Power Apps Home 🡪 Create blank app 🡪 canvas app (tablet)
2. On the left side, you now have the power automate icon where you can create a new flow

A screenshot of a computer

Description automatically generated

1. When you click create a flow, the power automate flow designer will open up. Construct a flow like the ff:

A screenshot of a computer

Description automatically generated

1. Set a flow name then click save
2. Add controls to send parameters from power app to your flow. Example:

sendEmailFlow.Run("johnreygoh@core360itservices.onmicrosoft.com","this is sent from power apps")

A screenshot of a computer

Description automatically generated

**Passing parameters from your flow back to your Power App**

1. Passing parameters back to your Power App is also possible. You can use the Respond to a PowerApp or flow action (from the PowerApps connector) for this:

A screenshot of a computer

Description automatically generated

1. This action will give you the same options as the Add an input from the PowerApps (V2) trigger, which allows you to pass various data types back to your Power App. You provide a name for the parameter and its value:

A screenshot of a computer

Description automatically generated

1. When you go back to your Power App and refresh your PowerFX formula, you will now see that the Run() command can be extended with the parameter you’ve just configured:

A screenshot of a computer

Description automatically generated

1. Running this command doesn’t get the response into your Power App yet, because there is no place to present it if you’re running it just from a button. To make sure you get the output you want, you need to put in into a variable of choice.
2. If you’re using a single screen Power App or don’t need the output to be used on another screen, you can use the UpdateContext() function for this:

UpdateContext({varResponse: 'Testing-PowerAppsbutton'.Run("Rik").response})

1. If you want to use the output on other screens as well, you can use the Set() function for this:

Set(

varResponse,

'Testing-PowerAppsbutton'.Run("Rik").response

)

1. After this, you can use this variable within your Power App to show the output (e.g. in a Label control):

A screenshot of a computer

Description automatically generated

1. Please note that the connection between a Power App and a flow will time out after 2 minutes. So if you’re having a complex flow that will run for more than 2 minutes, you won’t be getting any response back!