You

In Power apps, how can I go about making a flow that activates when a form is submitted and retrieves all the relevant details from that form, sending it to the Power app?

Copilot

Creating a flow in Power Apps that activates when a form is submitted and retrieves the details to send to the Power App is a great way to automate your processes! Here's a step-by-step guide to help you get started:

1. **Open Power Automate**: Go to Microsoft Power Automate and select **Create** > **Automated Flow**.
2. **Name Your Flow**: Give your flow a name or let Power Automate generate one for you.
3. **Choose Your Trigger**: In the search box under **Choose your flow's trigger**, type **forms**. Select **When a new response is submitted** and then click **Create**.
4. **Select Your Form**: Under **Form id**, click or tap in the search box to see a list of the forms you've created. Select the name of the form you want to use.
5. **Get Response Details**: Add a new step by clicking **New step**. In the search box under **Choose an action**, type **forms**. Select **Get response details**.
6. **Specify Form Details**: Under **Form id**, select the name of your form again. Under **Response id**, select **Response Id**.
7. **Add Conditions (Optional)**: If you need to perform different actions based on the form responses, you can add a **Condition** step. For example, you can use a condition to check a specific question's answer and perform different actions accordingly.
8. **Send Data to Power Apps**: To send the retrieved details to your Power App, you can use the **Create an item** action in SharePoint, the **Create a record** action in Dataverse, or any other action that integrates with your Power App.
9. **Save and Test**: Once you've configured your flow, save it and test it to ensure it works as expected.

Would you like more details on any specific step or help with a particular part of the process?

(1) Microsoft Support (2) Microsoft Learn

You

When I create the flow I get the following error:

Flow '16118c86-2e0d-4508-8d8d-9856a9a2f892' cannot be installed.
code: FlowCannotBeInstalled
Session ID: 4624783b-9431-4dda-a330-ffc3a808c7a7

Copilot

Thanks for sharing the error details! The **FlowCannotBeInstalled** error can occur for various reasons. Here are a few steps you can try to resolve it:

1. **Check Permissions**: Ensure that you have the necessary permissions to create flows in your environment. You might need to contact your admin if you're unsure.
2. **Connector Issues**: Sometimes, specific connectors might not be installed or enabled in your environment. Make sure all required connectors are properly set up.
3. **Recreate the Flow**: Try deleting the problematic flow and recreating it from scratch. Sometimes, simply recreating the flow can resolve unexpected issues.
4. **Check for Updates**: Ensure that your Power Automate and Power Apps are up to date. Sometimes, updates can fix bugs and issues.
5. **Review DLP Policies**: If your environment has Data Loss Prevention (DLP) policies, they might be blocking the flow. Check with your admin to see if any policies need to be adjusted.

If none of these steps work, you might want to reach out to Microsoft Support for further assistance.

Would you like to try any of these steps, or do you need help with something else?

(1) Microsoft Learn (2) Reddit (3) Cloud Ops Zone

You

If I can't start the flow with a form but I need to use a form to retrieve data, what else can I do?

Copilot

If you can't start the flow directly from a form, there are alternative ways to retrieve data from the form and use it in your flow. Here are a few methods you can consider:

1. **SharePoint Integration**:
    - Create a SharePoint list to store the form responses.
    - Use Power Apps to submit form data to the SharePoint list.
    - Set up a flow in Power Automate to trigger when a new item is added to the SharePoint list. This flow can then retrieve the form data and perform further actions.
2. **Microsoft Dataverse**:
    - Use Dataverse to store form responses.
    - Submit the form data to Dataverse using Power Apps.
    - Create a flow that triggers when a new row is added to the Dataverse table to process the form data.
3. **Manual Trigger**:
    - Create a Power App that includes the form.
    - Add a button to the Power App that users can click after submitting the form. This button can trigger the flow manually.
    - Configure the flow to retrieve the data submitted through the form and process it as needed.
4. **HTTP Request**:
    - Use a custom connector or an HTTP request to send the form data to a flow.
    - Trigger the flow using an HTTP request action that passes the form data as parameters.
    - Process the data in the flow accordingly.

Here's a quick example of how you can use the **SharePoint Integration** method:

1. **Create a SharePoint List**:
    - Set up a SharePoint list with columns that match the fields in your form.
2. **Submit Form Data to SharePoint**:
    - In Power Apps, configure the form to submit data to the SharePoint list when a user submits the form.
3. **Create a Flow**:
    - In Power Automate, create a new flow that triggers when a new item is created in the SharePoint list.
    - Add actions in the flow to retrieve the form data and send it to your Power App or other services as needed.

Would you like a more detailed guide on any of these methods?

You

(Now Monday, Nov 11)
How can i resolve the following error? [Uploaded Image Expired]

Copilot

The error you're encountering is a **502 Bad Gateway** error, indicating that the server did not receive a response from an upstream server. Here are some steps you can take to resolve this issue:

1. **Check the Upstream Server**: Ensure that the upstream server is running and accessible. This involves checking server logs, network connectivity, and server health.
2. **Retry the Request**: Sometimes, transient issues can cause a 502 error. Retrying the request after a short delay might resolve the issue.
3. **Check API Gateway Configuration**: Verify that the API gateway configuration is correct and that it is properly routing requests to the upstream server.
4. **Review Logs**: Check the logs of both the API gateway and the upstream server for any errors or warnings that might indicate the cause of the issue.
5. **Contact Support**: If the issue persists, consider contacting the support team for the API gateway or the upstream server for further assistance.
6. **Increase Timeout Settings**: Sometimes, increasing the timeout settings in your API gateway or in the settings where the request is being sent from can help.

Would you like more guidance on any of these steps or additional help with anything else?

You

given the context of my flow i was describig before, what can i do?#

Copilot

Given the context of your flow and the fact that you're facing issues with starting it directly from a form, you can consider an alternative approach to retrieve form data and process it. Here's a step-by-step guide using the **SharePoint Integration** method:

Step-by-Step Guide to Retrieve Form Data Using SharePoint Integration

1. Create a SharePoint List:

- Go to SharePoint and create a new list with columns that match the fields in your form. For example, if your form collects "Name," "Email," and "Feedback," create corresponding columns in your SharePoint list.

2. **Configure Power Apps to Submit Data to SharePoint**:
   - In Power Apps, create a form that users can fill out.
   - Use the SubmitForm function to submit the form data to the SharePoint list. Here's an example of how to set it up:
   - SubmitForm(FormName);
   - If(FormName.Error, Notify("There was an error submitting the form. Please try again.", NotificationType.Error), Notify("Form submitted successfully!", NotificationType.Success))
3. **Create a Flow in Power Automate**:
   - Open Power Automate and create a new flow.
   - Select the **When an item is created** trigger for SharePoint.
   - Choose the SharePoint site and list you created in Step 1.
4. **Retrieve and Process Form Data**:
   - Add actions in your flow to retrieve and process the form data. For example, you can send an email, update a database, or send data to another service.
   - Here's an example of how to send an email with the form data:
   - Add a new step: Send an email (V2)
   - To: <Your email address>
   - Subject: New Form Submission
   - Body: "A new form has been submitted with the following details:
 Name: @{triggerOutputs()?['body/Name']}
 Email: @{triggerOutputs()?['body/Email']}
 Feedback: @{triggerOutputs()?['body/Feedback']}"
5. **Save and Test**:
   - Save your flow and test it by submitting a form in Power Apps. Verify that the data is added to the SharePoint list and that the flow processes the data as expected.

Would you like more detailed assistance with any specific step or have any questions about this approach?

You

How can I extract the text from a word document for the sharepoint list in teh flow?

Copilot

To extract text from a Word document and store it in a SharePoint list using Power Automate, you can follow these steps:

1. **Upload the Word Document**: Ensure the Word document is uploaded to a SharePoint library.
2. **Create a Flow in Power Automate**:
   - Open Power Automate and create a new flow.
   - Select the **When a file is created (properties only)** trigger for SharePoint.
   - Choose the SharePoint site and library where the Word documents are stored.
3. **Get File Content**:
   - Add a new step by clicking **New step**.

    - In the search box, type **Get file content** and select the action.
    - Choose the **File Content** from the trigger.
4. **Extract Text from the Document**:
    - Add another step by clicking **New step**.
    - In the search box, type **Extract information from documents** and select the action.
    - Choose the document processing model you want to use. If you haven't created one yet, you can create a model in AI Builder to extract the necessary information from the Word document.
    - In the **Form** field, add **File Content** from the previous step.
5. **Store Extracted Data in SharePoint List**:
    - Add another step by clicking **New step**.
    - In the search box, type **Create an item** and select the action.
    - Choose the SharePoint site and list where you want to store the extracted data.
    - Map the extracted fields to the corresponding columns in the SharePoint list.
6. **Save and Test**:
    - Save your flow and test it by uploading a Word document to the SharePoint library. Verify that the data is correctly extracted and stored in the SharePoint list.

Would you like more detailed guidance on any of these steps or have any specific questions about the process?

References

1. process.st 2. learn.microsoft.com

You

How can I create an AI processing model for the Extract information from documents step?

Copilot

Creating an AI processing model in AI Builder to extract information from documents is a straightforward process. Here's a step-by-step guide to help you get started:

1. **Sign in to AI Builder**: Go to Power Apps or Power Automate and sign in.
2. **Navigate to AI Models**: On the left pane, select **AI hub**. Under **Discover an AI capability**, select **AI models**.
3. **Select Document Extraction**: Choose **Extract custom information from documents**.
4. **Create a Custom Model**: Click on **Create custom model**. A step-by-step wizard will guide you through the process.
5. **Define Information to Extract**: List all the data you want to extract from your document. This includes fields, tables, checkboxes, etc. For each field, provide a name and define the format (e.g., text, number, date).
6. **Group Documents by Collections**: If your documents have different layouts, group them by collections to ensure accurate extraction.
7. **Train the Model**: Select **Train** to start training your model. AI Builder will use your sample documents to learn how to extract the defined information.
8. **Test the Model**: After training, test the model by selecting **Quick test**. This will help you verify that the model is extracting the data correctly.
9. **Publish the Model**: Once you're satisfied with the test results, publish the model.

Would you like more detailed guidance on any of these steps or have any specific questions about the process?

References

learn.microsoft.com

You

How can I get the details and attributes associated with the last item after using apply to each in the flow?

Copilot

To get the details and attributes associated with the last item after using an **"Apply to each"** loop in Power Automate, you can use the following approach:

1. **Initialize a Variable**:
    - Before the "Apply to each" loop, create a variable to store the last item processed.
    - For example, add a **Initialize variable** action with a suitable name (e.g., LastItem). Set the variable type to **Object**.
2. **Set Variable Inside the Loop**:
    - Within the "Apply to each" loop, add a **Set variable** action to update the LastItem variable with the current item.
    - This way, each iteration will overwrite the variable with the latest item.
3. **Access the Last Item**:
    - After the "Apply to each" loop, you can access the LastItem variable to get the details and attributes of the last item processed.
    - Use the dynamic content from the LastItem variable as needed in subsequent actions.

Example Flow:

1. Initialize Variable:

Action: Initialize variable
Name: LastItem
Type: Object
Value: (leave empty)

2. Apply to Each:

Action: Apply to each
Input: (Your array items)

3. Set Variable Inside Loop:

Action: Set variable
Name: LastItem
Value: (The current item details)

4. Access Last Item Details:

- After the "Apply to each" loop, you can use the LastItem variable to access the details and attributes of the last item processed.

This method ensures that after the loop completes, the LastItem variable holds the data of the last item processed by the loop.

Would you like more detailed guidance on any specific part of this process?

You

How can I convert the date reetrive to a string?

Copilot

To convert a date to a string in Power Automate, you can use the **formatDateTime** function. This function allows you to format a date-time value as a string in a specific format. Here's how you can do it:

Step-by-Step Guide:

1. Add a Compose Action:
- Add a **Compose** action to your flow.
- In the **Inputs** field, use the **formatDateTime** function to convert the date to a string.

2. Use the formatDateTime Function:
- The syntax for the function is: formatDateTime(<date>, '<format>')
- For example, to convert the date to a string in the format "yyyy-MM-dd", you can use: *formatDateTime(triggerOutputs()?['body/<DateField>'], 'yyyy-MM-dd')*
- Replace <DateField> with the field that contains the date you want to convert.

3. Example:

formatDateTime(triggerOutputs()?['body/SubmissionDate'], 'yyyy-MM-dd')

Here's a more concrete example:

Example Flow:

1. Trigger: Your flow triggers (e.g., when a form is submitted or an item is created in SharePoint).

2. Compose Action:

Action: Compose

Inputs: formatDateTime(triggerOutputs()?['body/SubmissionDate'], 'yyyy-MM-dd')

This will convert the date in the SubmissionDate field to a string in the "yyyy-MM-dd" format.

Additional Formats:

You can use different format strings to get the desired date format. Here are some common examples:
- yyyy-MM-dd: 2024-11-11
- MM/dd/yyyy: 11/11/2024
- dd-MMM-yyyy: 11-Nov-2024
- dddd, MMMM dd, yyyy: Monday, November 11, 2024

Would you like further details on any specific part of this process?

You

How about getting the string reprsentation of a form choice (radiobox or chechbox)?

Copilot

To get the string representation of a form choice (radio button or checkbox) in Power Automate, you can use dynamic content and expressions. Here's how you can achieve this:

Step-by-Step Guide:

1. **Retrieve the Form Response**:
   - Use the **Get response details** action to retrieve the form response.
   - Ensure that your form choices (radio button or checkbox) are included in the response details.
2. **Extract the Choice Value**:
   - In the subsequent actions, use the dynamic content to extract the choice value.
   - If the choice is a single selection (radio button), you can directly use the dynamic content.
   - If the choice is a multi-selection (checkbox), you'll need to convert it to a string representation.

Example Flow:

Single Selection (Radio Button):

1. Get Response Details:

Action: Get response details

Form ID: (Your Form ID)

Response ID: (Dynamic content from trigger)

2. Compose Action:

Action: Compose

Inputs: @{outputs('Get_response_details')?['body/YourRadioButtonField']}

Multi-Selection (Checkbox):

1. Get Response Details:

Action: Get response details
Form ID: (Your Form ID)
Response ID: (Dynamic content from trigger)

2. Join Function to Convert Array to String:

- Add a **Compose** action to convert the array of selected options into a string.
  Action: Compose
  Inputs: join(outputs('Get_response_details')?['body/YourCheckboxField'], ', ')

Explanation:

- outputs('Get_response_details')?['body/YourRadioButtonField']: Retrieves the value of the radio button choice.
- join(outputs('Get_response_details')?['body/YourCheckboxField'], ', '): Converts the array of selected checkbox options into a single string, separated by commas.

These steps will allow you to get the string representation of the selected options from your form.
Would you like more detailed guidance on any of these steps, or do you have any other questions?

You

This is a screenshot of the various parts of my flow, it's still not working and throwimg the same error as before when I try to output the return variable as text in my power app:

SharePointLister.Run failed: {
"error": {
"code": 502,
"source": "0713ee51-a2c3-420b-bf1f-61553c866fc1.08.common.uk.azure-apihub.net",
"clientRequestId": "1fc8336d-bea6-4921-893b-a293efa6850a",
"message": "BadGateway",
"innerError": {
"error": {
"code": "NoResponse",
"message": "The server did not receive a response from an upstream server. Request tracking id '08584702791639505548067053224CU10'."

[Uploaded Screnshot Expired]

Copilot

Thanks for sharing the screenshot! It looks like you're encountering a **502 Bad Gateway** error, which typically indicates that the server didn't receive a response from an upstream server. Here are some steps to troubleshoot and resolve this issue:

1. **Check Network Connectivity**: Ensure that both Power Apps and Power Automate have a stable internet connection. If necessary, try restarting your network devices or contacting your IT department for assistance.
2. **Verify Upstream Server**: Make sure the upstream server (e.g., SharePoint) is running and accessible. Check server logs and network connectivity.
3. **Retry the Flow**: Sometimes, transient issues can cause a 502 error. Retry running the flow to see if the problem persists.
4. **Review Flow Configuration**: Double-check the flow configuration to ensure all connections and permissions are correctly set up.
5. **Increase Timeout Settings**: If the issue is related to timeouts, consider increasing the timeout settings in your flow.
6. **Contact Support**: If the issue persists, consider contacting Microsoft support for further assistance.

Would you like more detailed guidance on any of these steps, or do you have any other questions?

References

1. answers.microsoft.com 2. devcodef1.com

You

How can I do it based on my flow [Uploaded Screenshot Expired]

Copilot

Thanks for sharing the details of your flow! Let's troubleshoot the issue step-by-step, focusing on the elements in your flow and addressing the **502 Bad Gateway** error.

Overview of Your Flow Steps:

1. **Power Apps Trigger**: Initiates the flow.
2. **When a new response is submitted**: Triggered by a form submission.
3. **Initialize Variable**: Initializing variables to store information.
4. **Apply to Each**: Loop through each item in the list of responses.
5. **Get Response Details**: Retrieve details of each form response.
6. **Create Item**: Create a new item in a SharePoint list.
7. **Set Variable**: Update variables as needed.
8. **Get Item**: Retrieve the last created item from SharePoint.
9. **Condition**: Check certain conditions and set variables accordingly.
10. **Respond to a Power App**: Send a response back to Power Apps.

Steps to Address the Error:

1. **Check the SharePoint Connection**:
   - Ensure that the SharePoint connectors are properly configured and that you have the necessary permissions to access the SharePoint list.
   - Verify that the SharePoint site and list names are correct.
2. **Increase Timeout**:
   - The error might be due to a timeout issue. Increase the timeout settings for the **Create Item** and **Get Item** actions.
3. **Break Down Complex Steps**:
   - Simplify the flow by breaking down complex steps. This can help identify where the issue is occurring.
   - For example, you can separate the **Create Item** and **Get Item** actions into separate flows to see if the error persists.
4. **Test Each Component**:
   - Test each action individually to ensure they work as expected.
   - Start by testing the **Create Item** action to ensure it successfully adds an item to the SharePoint list.
   - Next, test the **Get Item** action to ensure it retrieves the item correctly.
5. **Review and Optimize Conditions**:
   - Ensure that the conditions in your flow are correctly set up and that they don't cause any issues.
   - Make sure that all variables used in conditions are properly initialized and updated.
6. **Log and Debug**:
   - Add logging actions (e.g., **Compose** or **Append to array variable**) at different points in the flow to capture the state of variables and actions.
   - This will help you identify where the flow is failing.
7. **Retry Logic**:
   - Implement retry logic for the actions that are prone to errors, such as **Create Item** and **Get Item**.
   - You can configure the retry policy in the settings of these actions.

Example Changes to Your Flow:

1. Increase Timeout for Create Item:
- Go to the **Settings** of the **Create Item** action and increase the **Timeout** value.

2. Add Logging Actions:
- Add **Compose** actions after critical steps to log the output. For example:
  Action: Compose
  Inputs: outputs('Create_Item')?['body']

3. Retry Logic:
- Configure retry settings for actions:
  Go to Settings of the action > Retry policy > Set to `Exponential` or `Fixed` with a specified retry count and interval.

Would you like more detailed guidance on any specific step, or do you have any additional questions about your flow?

You

Say I have an array with 1 item in it returned from the sharepoint list, how can I store that one item as a string?

Copilot

To store an item from an array as a string in Power Automate, you can use a **Compose** action or a **Set variable** action. Here's how you can do it:

Using Compose Action

1. **Initialize the Array**:
   - Ensure you have an array with at least one item. For example, let's assume the array is named MyArray.
2. **Add Compose Action**:
   - Add a **Compose** action to extract the item from the array and convert it to a string.
   - Use an expression to get the first item in the array.

Example:

Action: Compose
Inputs: first(outputs('YourPreviousAction')?['body/MyArray'])

Using Set Variable Action

1. **Initialize Variable**:
   - Initialize a string variable to store the item.
2. **Set Variable Inside Apply to Each** (if inside a loop):
   - Use the **Set variable** action to update the string variable with the array item.
   - Ensure the **Apply to each** action processes only the first item if the array has multiple items.

Example:
1. Initialize Variable:

Action: Initialize variable
Name: MyStringVariable
Type: String
Value: (leave empty)

2. Set Variable:

Action: Set variable
Name: MyStringVariable
Value: first(outputs('YourPreviousAction')?['body/MyArray'])

In these examples:

- Replace YourPreviousAction with the action name that produces the array.
- Replace MyArray with the actual name of your array.

This will convert the first item in your array to a string and store it in the MyStringVariable variable.
Would you like more detailed guidance on any specific step or have any other questions?

You

How about if the array has multiple items, how can I include a string of each one, separated by commas?

Copilot

To concatenate all items in an array into a single string with each item separated by commas, you can use the **join** function in Power Automate. Here's how you can do it:

Step-by-Step Guide:

1. **Get the Array**:
   - Ensure you have an array with multiple items. Let's assume the array is named MyArray.
2. **Use the Join Function**:
   - Add a **Compose** action to convert the array into a single string with items separated by commas.

Example Flow:

1. **Add a Compose Action**:
    - In Power Automate, add a **Compose** action.
    - Use the **join** function to concatenate the array items.

Example:

Action: Compose
Inputs: join(outputs('YourPreviousAction')?['body/MyArray'], ', ')

In this example:

- Replace YourPreviousAction with the action name that produces the array.
- Replace MyArray with the actual name of your array.

This will result in a single string with all items from the array separated by commas.

Explanation:

- *join(outputs('YourPreviousAction')?['body/MyArray'], ', ')*: This expression joins all the items in the array MyArray into a single string, separated by commas.

Full Example in a Flow:

1. **Trigger**: Your flow triggers (e.g., when a new item is created in SharePoint or a form is submitted).
2. **Get Array**: Retrieve the array from a previous action.
3. **Compose Action to Join Array**:
    Action: Compose
    Inputs: join(outputs('YourPreviousAction')?['body/MyArray'], ', ')

This will convert the array items into a single string separated by commas.
Would you like further details on any specific part of this process or help with something else?

You

How can I show a form question only if the user selects other on the previosu question and hide it otherwise>?

Copilot

To show a form question only if the user selects "Other" on the previous question and hide it otherwise, you can use conditional logic in Power Apps. Here's how you can achieve this:

Step-by-Step Guide:

1. **Create Your Form**:
    - Create a form in Power Apps with the necessary fields.
2. **Add Your Questions**:
    - Add the main question with options (including "Other").
    - Add the question you want to show conditionally based on the user's selection.
3. **Add a Drop-down or Radio Button Control**:
    - For the main question, use a **Drop-down** or **Radio Button** control.
    - Set the items property of this control to the options you want (e.g., ["Option 1", "Option 2", "Other"]).
4. **Set Conditional Visibility**:
    - Select the question you want to show conditionally.
    - Go to the properties pane and find the **Visible** property.
    - Set the **Visible** property to the following formula:
        If(Dropdown1.Selected.Value = "Other", true, false)
    - Replace Dropdown1 with the name of your Drop-down or Radio Button control.

Example:

1. **Add a Drop-down for Main Question**:
    - Insert a **Drop-down** control and set its **Items** property to:
        ["Option 1", "Option 2", "Other"]
    - Name this control Dropdown_MainQuestion.
2. **Add a Text Input for the Conditional Question**:
    - Insert a **Text Input** control for the "Other" option.
    - Name this control TextInput_Other.
3. **Set the Visible Property**:
    - Select the TextInput_Other control.
    - Set its **Visible** property to:
        If(Dropdown_MainQuestion.Selected.Value = "Other", true, false)

This will ensure that the TextInput_Other control is only visible when the user selects "Other" from the Dropdown_MainQuestion control.

Would you like more detailed guidance on any specific step, or do you have any other questions?

You

Not in PowerApps but SharePoint?

Copilot

Got it! To make a form question appear only if the user selects "Other" on the previous question in SharePoint, you can use a combination of SharePoint lists and Power Automate. Here's how you can achieve this:

Step-by-Step Guide:

1. **Create SharePoint Lists**:
    - **Main List**: This list will store the initial form responses, including the selection of "Other".
    - **Conditional List**: This list will store the additional details if "Other" is selected.
2. **Create Columns in Main List**:
    - Add columns for each question in the form, including a choice column for the main question with options (e.g., "Option 1", "Option 2", "Other").
3. **Create Columns in Conditional List**:
    - Add columns for the additional details you want to capture if "Other" is selected.
4. **Create a Power Automate Flow**:
    - Create a flow that triggers when a new item is created in the Main List.
    - Add a condition to check if the "Other" option was selected.
    - If "Other" is selected, prompt the user to fill in additional details and store them in the Conditional List.

Detailed Flow Steps:

1. Trigger:
- Trigger the flow when a new item is created in the Main List.

2. Condition:
- Add a **Condition** action to check if the main question's response is "Other".
  Condition: @equals(triggerOutputs()?['body/MainQuestionField'], 'Other')

3. If Yes - Send Form Link:
    - If the condition is true (i.e., "Other" is selected), use the **Send an email (V2)** action to send a link to the user to fill in the additional form.

4. Form Submission to Conditional List:
    - Create a secondary form (Microsoft Forms) for additional details and set it up to submit responses to the Conditional List.

5. Add New Item to Conditional List:
    - Use the **Create item** action in Power Automate to add a new item to the Conditional List with the additional details provided by the user.

Example Flow:

1. Trigger:
When a new item is created in the Main List

2. Condition:
@equals(triggerOutputs()?['body/MainQuestionField'], 'Other')

3. If Yes:
- **Send an email (V2)**:
  To: UserEmail
  Subject: Additional Information Required
  Body: Please fill in the additional details using the following link: [Link to Microsoft Forms]

4. Form Submission to Conditional List:
    - Create item in the Conditional List using the responses from the secondary form.

This setup ensures that when a user selects "Other" in the main question, they will receive a prompt to provide additional details, and these details will be stored in a separate SharePoint list.

Would you like further details on any specific step, or do you have any other questions?

You

How I automatically fill out a column in teh sharepoint list when a new item is added, I have a request ID column and an item is added when a user fills out teh form but reequest ID isn't included in teh form ofc and tehn it is blank when the item is created based on the form reponse, it should be in teh format SAPCR#000004 where teh number is incremented with each new entry starting from SAPCR#000007

Copilot

To automatically fill out a "Request ID" column in your SharePoint list when a new item is added, you can use a Power Automate flow to generate the ID in the format **SAPCR#000004** and increment it with each new entry. Here's how to set this up:

Step-by-Step Guide:

1. **Initialize a Variable for Incrementing**:
   - Create a variable in your flow to store the current highest Request ID number.
2. **Get the Highest Request ID**:
   - Use the **Get items** action to retrieve items from the SharePoint list.
   - Sort the items in descending order based on the Request ID to get the latest Request ID.
3. **Increment the Request ID**:
   - Parse the highest Request ID to extract the number and increment it by one.
4. **Generate the New Request ID**:
   - Format the new Request ID in the required format (e.g., SAPCR#000007).
5. **Update the SharePoint Item**:
   - Use the **Update item** action to update the newly created item with the generated Request ID.

Detailed Steps:

1. **Initialize Variable**:
- Add an **Initialize variable** action to store the current highest Request ID.
  Action: Initialize variable
  Name: HighestRequestID
  Type: Integer
  Value: 6 (Starting from SAPCR#000007 which is increment 6 from your format)

Get Items from SharePoint List:
- Use the **Get items** action to retrieve the list items.
  Action: Get items
  Site Address: Your SharePoint site
  List Name: Your SharePoint list
  Order By: RequestID desc
  Top Count: 1 (To get the highest Request ID)

Parse and Increment the Request ID:
- Use a **Compose** action to parse the highest Request ID and increment it.
  Action: Compose
  Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)

Generate the New Request ID:
- Use a **Compose** action to format the new Request ID.
  Action: Compose
  Inputs: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))

Update the SharePoint Item:
- Use the **Update item** action to update the newly created item with the new Request ID.
  Action: Update item
  Site Address: Your SharePoint site
  List Name: Your SharePoint list
  Id: ID of the newly created item
  RequestID: outputs('Compose')

Example Flow:

1. **Trigger**:
   - When an item is created in the SharePoint list.
2. **Initialize Variable**:
   Action: Initialize variable
   Name: HighestRequestID
   Type: Integer
   Value: 6
3. **Get Items**:
   Action: Get items
   Site Address: Your SharePoint site
   List Name: Your SharePoint list
   Order By: RequestID desc
   Top Count: 1
4. **Parse and Increment**:
   Action: Compose
   Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)
5. **Generate Request ID**:
   Action: Compose
   Inputs: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))
6. **Update Item**:
   Action: Update item
   Site Address: Your SharePoint site
   List Name: Your SharePoint list
   Id: ID of the newly created item
   RequestID: outputs('Compose')

This setup ensures that each new item in the SharePoint list gets a unique Request ID in the specified format, incrementing from the last Request ID.
Would you like more detailed guidance on any specific step, or do you have any other questions?

You

How can I create an ODATA orderBy query for this step: Sort the items in descending order based on the Request ID to get the latest Request ID.

Copilot

You can use the OData query options in Power Automate to sort the items in descending order based on the Request ID. The OData orderBy query parameter is used for this purpose. Here's how to create the OData query:

Step-by-Step Guide:

1. **Get Items Action**:
   - In your flow, add or configure the **Get items** action to retrieve items from your SharePoint list.
2. **Specify the OData Query**:
   - In the **Get items** action, locate the **Order By** field.
   - To sort items by the RequestID column in descending order, enter the following query:
     RequestID desc

Example Flow Configuration:

1. **Get Items**:
   - **Site Address**: Your SharePoint site
   - **List Name**: Your SharePoint list
   - **Order By**: RequestID desc
   - **Top Count**: 1 (to get the latest item)

Full Example:

Action: Get items
Site Address: https://yoursharepointsite.sharepoint.com/sites/yoursite

List Name: YourListName
Order By: RequestID desc
Top Count: 1

Explanation:

- **Order By**: RequestID desc sorts the items by the RequestID column in descending order, ensuring that the item with the highest Request ID is retrieved first.
- **Top Count**: 1 ensures that only the latest item (with the highest Request ID) is retrieved.

This configuration will allow you to get the latest item based on the RequestID and proceed with incrementing it for new entries.
Would you like more detailed guidance on any specific step or have any other questions?

You

How about if the top ID is already "SAPCR#000006", what would be the flow steps then?

Copilot

It sounds like you're working on a system where IDs are used to track something, and the current top ID is "SAPCR#000006." Here's a general flow for continuing from that point:

1. **ID Generation**: When a new entry needs an ID, increment the existing top ID. So, the next ID would be "SAPCR#000007".
2. **Check for Uniqueness**: Ensure this new ID is not duplicated. Most systems will have checks in place to confirm that the new ID is unique.
3. **Assign ID**: Assign the generated ID to the new entry.
4. **Store Entry**: Save the new entry with its assigned ID to your database or tracking system.
5. **Update Top ID**: Update the system's reference for the top ID to "SAPCR#000007" for future increments.

Make sure your ID generation script or function is robust to handle potential edge cases, such as simultaneous requests. Some systems use transaction locks or unique constraints in their database to manage this.

Does this align with your needs or is there a specific system you're working with?

You

What would be the power automate flow then?

Copilot

To create a Power Automate flow for generating and assigning IDs, you can follow these steps:

1. **Create a New Flow**: Open Power Automate and create a new flow. Choose "Manually trigger a flow" under "Choose how to trigger this flow".
2. **Add Compose Action**: Add a "Compose" action to generate the new ID. Use the expression guid() to generate a globally unique identifier (GUID).
3. **Check for Uniqueness**: Add a condition to check if the generated ID already exists in your database or list. You can use the "Get items" action to query your database or list.
4. **Assign ID**: If the ID is unique, proceed to assign it to the new entry. You can use the "Create item" or "Update item" action to save the new entry with the assigned ID.
5. **Update Top ID**: If the ID is assigned successfully, update the reference for the top ID. You can use the "Update item" action to set the new top ID.
6. **Store Entry**: Save the new entry with its assigned ID to your database or tracking system.
7. **Notify User**: Optionally, add a step to notify the user that the ID has been successfully assigned.

Here's a simplified example of what the flow might look like:

1. **Trigger**: Manually trigger the flow.
2. **Compose**: `guid()` to generate a new ID.
3. **Condition**: Check if the ID exists in the database.
4. **If Yes**: Go to the next step.
5. **If No**: Assign the ID to the new entry and update the top ID.
6. **Store Entry**: Save the new entry with the assigned ID.
7. **Notify User**: Send a notification about the successful ID assignment.

Does this help with what you're trying to achieve?

References

1. process.st.com 2. learn.microsoft.com 3. powertechtips.com 4. techcommunity.microsoft.com

You

I'm saying how would this flow be amended if my top ID was "SAPCR#000006" to begin with:
Example Flow:
Trigger: When an item is created in the SharePoint list.
Initialize Variable: plaintext

Action: Initialize variable
Name: HighestRequestID
Type: Integer
Value: 6

Get Items: plaintext
Action: Get items
Site Address: Your SharePoint site
List Name: Your SharePoint list
Order By: RequestID desc
Top Count: 1

Parse and Increment: plaintext
Action: Compose
Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)

Generate Request ID: plaintext
Action: Compose
Inputs: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))

Update Item: plaintext
Action: Update item
Site Address: Your SharePoint site
List Name: Your SharePoint list
Id: ID of the newly created item
RequestID: outputs('Compose')

This setup ensures that each new item in the SharePoint list gets a unique Request ID in the specified format, incrementing from the last Request ID.

Copilot

Got it! Thanks for sharing your flow. If your top ID starts at "SAPCR#000006," you can follow these steps to amend your Power Automate flow:

1. **Trigger:**
    - When an item is created in the SharePoint list.
2. **Initialize Variable:**

    - **Action**: Initialize variable
    - **Name**: HighestRequestID
    - **Type**: Integer
    - **Value**: 6 (This is your initial top ID value)
3. **Get Items:**
    - **Action**: Get items
    - **Site Address**: Your SharePoint site
    - **List Name**: Your SharePoint list
    - **Order By**: RequestID desc
    - **Top Count**: 1 (This will get the item with the highest RequestID)
4. **Parse and Increment:**
    - **Action**: Compose
    - **Inputs**: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)
    - This step extracts the numeric part of the highest RequestID and increments it by 1.
5. **Generate Request ID:**
    - **Action**: Compose
    - **Inputs**: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))
    - This step generates the new RequestID by combining "SAPCR#" with the incremented number, padded to 6 digits.
6. **Update Item:**
    - **Action**: Update item
    - **Site Address**: Your SharePoint site
    - **List Name**: Your SharePoint list
    - **Id**: ID of the newly created item
    - **RequestID**: outputs('Compose')
    - This step assigns the new RequestID to the newly created item.

Here's how your amended flow would look:

1. **Trigger:** When an item is created in the SharePoint list.

2. **Initialize Variable:**
   - Action: Initialize variable
   - Name: HighestRequestID
   - Type: Integer
   - Value: 6

3. **Get Items:**
   - Action: Get items
   - Site Address: Your SharePoint site
   - List Name: Your SharePoint list
   - Order By: RequestID desc
   - Top Count: 1

4. **Parse and Increment:**
   - Action: Compose
   - Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)

5. **Generate Request ID:**
   - Action: Compose
   - Inputs: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))

6. **Update Item:**
   - Action: Update item
   - Site Address: Your SharePoint site
   - List Name: Your SharePoint list
   - Id: ID of the newly created item
   - RequestID: outputs('Compose')

This setup ensures that each new item in the SharePoint list gets a unique RequestID in the specified format, incrementing from the last RequestID.

Does this help with what you're trying to achieve? Anything else you'd like to know?

You

(Now Wednesday, Nov 13)
In power apps, I have a gallery called Gallery1 that is connected to a sharepoint list, how can I retrieve the data fields from that gallery to use lesewhere in my app?

Copilot

To retrieve data fields from a gallery in Power Apps and use them elsewhere in your app, you can follow these steps:

Step-by-Step Guide

1. **Reference the Gallery**:
   - You can reference the gallery items using *Gallery1.Selected*. This will give you access to the currently selected item in the gallery.
2. **Access Specific Fields**:
   - To access specific fields from the selected item, use the dot notation. For example, if you want to retrieve the Title field from the selected item, you would use *Gallery1.Selected.Title*.
3. **Store Data in Variables**:
   - To use the retrieved data elsewhere, store it in a variable. You can do this using the **Set** function.

Example

1. Select an Item in the Gallery:

- Make sure your gallery is set up to allow item selection.

2. Store Selected Item Data in Variables:

- When an item is selected, store the necessary fields in variables using a button or the **OnSelect** property of the gallery.

  Set(SelectedTitle, Gallery1.Selected.Title);
  Set(SelectedID, Gallery1.Selected.ID);
  Set(SelectedDescription, Gallery1.Selected.Description)

3. Use the Variables Elsewhere:

- You can now use these variables anywhere in your app. For example, display the selected item's title in a label:

  Text: SelectedTitle

Detailed Steps

1. OnSelect Property of Gallery:

- Set the **OnSelect** property of the gallery to capture the selected item's fields.

  OnSelect: Set(SelectedTitle, ThisItem.Title)

2. Add Controls to Display Data:

- Add a label to display the *Title* field:

  Text: SelectedTitle

3. Use Data in Forms or Other Controls:

- If you have a form that should display the selected item's data, set the form's *Item* property to the selected item:

  Item: Gallery1.Selected

Additional Example: Using Data in a Different Screen

1. Navigate to a New Screen:

- Add a button to navigate to a new screen and store data:

  OnSelect: Navigate(Screen2); Set(SelectedTitle, Gallery1.Selected.Title)

2. Display Data in the New Screen:

- In the new screen, add a label to display the *SelectedTitle* variable:

  Text: SelectedTitle

These steps will help you retrieve and use data from your gallery in different parts of your Power App.
Would you like more detailed guidance on any specific step, or do you have any other questions?

You

How about to display all the items in teh gallery?

Copilot

To display all the items in a gallery in Power Apps, you'll need to ensure that the gallery is correctly connected to your data source, and then configure the gallery to display the data fields you want. Here's how to do it:

Step-by-Step Guide:

1. Set Up the Gallery:

- Ensure your gallery (*Gallery1*) is connected to your SharePoint list or other data source.
- In the **Items** property of the gallery, set it to the data source. For example:
  Items: YourSharePointList

2. Display Fields in the Gallery:

- Inside the gallery, you can add controls (like labels, text boxes) to display the fields from your data source.
- For each control, set the **Text** property to the corresponding field. For example, if you want to display the *Title* field from your SharePoint list:
  Text: ThisItem.Title

Example Setup:

1. Connect the Gallery to the Data Source:

- Select *Gallery1*.
- Set the **Items** property of the gallery:
  Items: YourSharePointList

2. Add Controls to Display Fields:

- Add a label inside the gallery template to display the *Title* field:
  Text: ThisItem.Title
- Similarly, add other controls (like labels, text boxes) for additional fields:
  Text: ThisItem.Description
  Text: ThisItem.CreatedDate

Example Complete Configuration:

1. Gallery Items Property:
Items: YourSharePointList

2. Label for Title:
Text: ThisItem.Title

3. Label for Description:
Text: ThisItem.Description

4. Label for Created Date:
Text: ThisItem.CreatedDate

This configuration will ensure that all items from your SharePoint list are displayed in Gallery1, with each field properly shown in the gallery items.

Would you like more detailed guidance on any specific step, or do you have any other questions?

You

Some of my data types that I want to display are stored as records (e.g. result of radio button selection), how can I display this as text?

Copilot

To display data types stored as records (such as the result of a radio button selection) as text in Power Apps, you can use the *ThisRecord* property and extract the specific value you want to display. Here's how to do it:

Step-by-Step Guide:

1. **Identify the Record Field**:
   - Determine which field in your gallery items contains the record data. For example, let's say your gallery is connected to a SharePoint list, and the column storing the radio button selection results is *Selection*.
2. **Add a Label to the Gallery**:
   - Inside the gallery template, add a label to display the value.
3. **Set the Label Text Property**:
   - Set the *Text* property of the label to extract and display the value from the record.

Example:

1. Gallery Items Property:

- Ensure your gallery (*Gallery1*) is connected to your SharePoint list.

  Items: YourSharePointList

2. Add a Label to the Gallery:

- Insert a label inside the gallery template to display the value from the *Selection* field.

3. Extract the Record Value:

- Assuming the *Selection* field stores a record with a value attribute, set the Text property of the label to:

  Text: ThisItem.Selection.Value

Detailed Example:

1. Connect the Gallery to the Data Source:

Items: YourSharePointList

2. Add a Label:

- Insert a label inside the gallery template.

3. Set the Text Property:

- Assuming *Selection* is the field storing the radio button result, and it is a record with a *Value* property:

  Text: ThisItem.Selection.Value

General Approach:

- If the record has nested attributes, use the dot notation to access the specific value. For example:

  Text: ThisItem.Selection.RecordField.Value

This setup will ensure that the text representation of your record data is displayed correctly in the gallery.

Would you like further details on any specific step, or do you have any other questions?

You

How can I write an if statement in the text label that shows ThisItem.'Other Requestor Department' if ThisItem.'Requestor Department'.Value is "Other" and ThisItem.'Requestor Department'.Value otherwise?

Copilot

To achieve this, you can use the If function in the Text property of the label. This function will allow you to conditionally display text based on the value of ThisItem.'Requestor Department'.Value. Here's how you can write the If statement:

If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value)

Explanation:

- ThisItem.'Requestor Department'.Value = "Other": This condition checks if the value of the Requestor Department field is "Other".
- ThisItem.'Other Requestor Department': This is the text that will be displayed if the condition is true (i.e., if the value is "Other").
- ThisItem.'Requestor Department'.Value: This is the text that will be displayed if the condition is false (i.e., if the value is anything other than "Other").

Step-by-Step Guide:

1. **Select the Label**:
   - Click on the label in your gallery where you want to display the text conditionally.
2. **Set the Text Property**:
   - In the **Text** property of the label, enter the If statement:

     Text: If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value)

Example:

1. **Gallery**: Assume you have a gallery named Gallery1 that displays items from a SharePoint list.
2. **Fields**: The list includes the fields 'Requestor Department' and 'Other Requestor Department'.
3. **Label in Gallery**: Add a label inside the gallery to display the department information conditionally.
4. **Set Text Property**:

   Text: If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value)

This setup will ensure that the label displays *'Other Requestor Department'* if *'Requestor Department'.Value* is "Other", and otherwise displays the value of *'Requestor Department'.Value*.
Would you like further details on any specific step, or do you have any other questions?

You

The actual text I want to display should be in this format: *Concatenate("Requestor Department: ",RequestorDep)*

Copilot

You can concatenate the text in the desired format using the *Concatenate* function in Power Apps. Here's how you can do it, incorporating the conditional logic:

Step-by-Step Guide:

1. **Select the Label**:
   - Click on the label in your gallery where you want to display the concatenated text.
2. **Set the Text Property**:
   - Use the *Concatenate* function along with the *If* function to format the text as required.

Example:

Text: Concatenate("Requestor Department: ", If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value))

Explanation:

- *Concatenate("Requestor Department: ", ...)*: This concatenates the string "Requestor Department: " with the value returned by the *If* function.
- *If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value)*: This checks if the *Requestor Department* value is "Other" and returns the appropriate value.

Full Example Configuration:

1. **Gallery**: Ensure your gallery (Gallery1) is connected to your SharePoint list.
   Items: YourSharePointList
2. **Add a Label**:
   - Insert a label inside the gallery template.
3. **Set the Text Property**:
   Text: Concatenate("Requestor Department: ", If(ThisItem.'Requestor Department'.Value = "Other", ThisItem.'Other Requestor Department', ThisItem.'Requestor Department'.Value))

This setup will display the text in the desired format, showing "Requestor Department: [value]" with the correct department value based on the user's selection.
Would you like further details on any specific step, or do you have any other questions?

You

Also I've got teh Request date as a String (ThisItem.'Request Date') but I want to display it in a different format (a current example is: 11/5/2024), however I want to display it either as DD/MM/YYYY or Day, Month, Year (using that example, either like 5/11/2024 or 5 November, 2024), how can I do this?

Copilot

You can format the date string in Power Apps using the *DateValue*, *Text*, and *Date* functions. Here's how you can do it:

Step-by-Step Guide:

1. **Convert the String to a Date**:
   - Use the *DateValue* function to convert the string to a date.
2. **Format the Date**:
   - Use the *Text* function to format the date as DD/MM/YYYY or Day, Month, Year.

Example:

1. **Convert String to Date**:
   - Assuming the string is in the format MM/DD/YYYY, use the *DateValue* function:
     DateValue(ThisItem.'Request Date', "en-US")
2. **Format as DD/MM/YYYY**:
   - Use the *Text* function to format the date:
     Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]dd/mm/yyyy")

3. **Format as Day, Month, Year**:
   - Use the *Text* function to format the date:
     Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]d mmmm, yyyy")

Detailed Steps:
1. **Add a Label to Display the Formatted Date**:
   - Insert a label in your Power App.
2. **Set the Text Property for DD/MM/YYYY**:
   Text: Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]dd/mm/yyyy")
3. **Set the Text Property for Day, Month, Year**:
   Text: Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]d mmmm, yyyy")

Explanation:
- **DateValue(ThisItem.'Request Date', "en-US")**: Converts the string to a date assuming the format MM/DD/YYYY.
- **Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]dd/mm/yyyy")**: Formats the date as DD/MM/YYYY.
- **Text(DateValue(ThisItem.'Request Date', "en-US"), "[$-en-GB]d mmmm, yyyy")**: Formats the date as Day, Month, Year.

This approach will help you display the *Request Date* in the desired format.
Would you like more detailed guidance on any specific step, or do you have any other questions?

You

How about to concatenate all teh values of a table (with a comma inbetween) refernced as ThisItem.'Request Departments'

Copilot

To concatenate all the values in a table (or collection) and separate them with commas in Power Apps, you can use the *Concat* function. This function concatenates the results of a formula applied to each record of a table. Here's how you can achieve this with your *ThisItem.'Request Departments'* table:

Step-by-Step Guide
1. **Use the Concat Function**:
   - The *Concat* function allows you to concatenate values from a table.
   - The syntax is: *Concat(Table, Formula, Separator)*
2. **Set the Text Property**:
   - Use the *Concat* function in the *Text* property of a label to display the concatenated values.

Example:
Assume *ThisItem.'Request Departments'* is the table containing the department values.
1. **Set the Text Property of a Label**:
   Text: Concat(ThisItem.'Request Departments', Value, ", ")

Explanation:
- *ThisItem.'Request Departments*': The table you want to concatenate values from.
- *Value*: The field within each record of the table that you want to concatenate. Replace Value with the actual field name if it is different.
- *", "*: The separator to use between concatenated values (in this case, a comma and a space).

Detailed Steps:
1. **Select the Label**:
   - Click on the label where you want to display the concatenated values.
2. **Set the Text Property**:
   - Enter the following formula:
     Text: Concat(ThisItem.'Request Departments', Value, ", ")
   - Replace *Value* with the actual field name within each record that you want to concatenate.

This setup will concatenate all the values in the *ThisItem.'Request Departments'* table, separating them with commas.
Would you like more detailed guidance on any specific step, or do you have any other questions?

You

How about keeping the concat but if there is the value "Other" prsent within the table, then other is replaced by the value refernced by ThisItem.'Other Request Departments'

Copilot

To achieve this, you can use the *Concat*, *If*, and *ForAll* functions in Power Apps to conditionally replace the value "Other" with *ThisItem.'Other Request Departments'* while concatenating the values from the *ThisItem.'Request Departments'* table. Here's how you can do it:

Step-by-Step Guide

1. **Use ForAll to Iterate Over the Table**:
   - Use the *ForAll* function to iterate over each item in the table.
   - Use the *If* function to check if the value is "Other" and replace it accordingly.
2. **Concatenate the Values**:
   - Use the *Concat* function to concatenate the values with a comma separator.

Example Formula:

```
Text: Concat(
   ForAll(
      ThisItem.'Request Departments',
      If(
         Value = "Other",
         ThisItem.'Other Request Departments',
         Value
      )
   ),
   ", "
)
```

Explanation:

- *ForAll(ThisItem.'Request Departments', ...)*: Iterates over each record in the Request Departments table.
- *If(Value = "Other", ThisItem.'Other Request Departments', Value)*: Checks if the value is "Other" and replaces it with the value from Other Request Departments, otherwise keeps the original value.
- *Concat(..., ", ")*: Concatenates the resulting values with a comma and a space as separators.

Detailed Steps:

1. **Select the Label**:
   - Click on the label where you want to display the concatenated values.
2. **Set the Text Property**:
   Enter the following formula:

```
Text: Concat(
   ForAll(
      ThisItem.'Request Departments',
      If(
         Value = "Other",
         ThisItem.'Other Request Departments',
         Value
      )
   ),
   ", "
)
```

This will concatenate all the values in the *Request Departments* table, replacing "Other" with the value from *Other Request Departments*, and separate them with commas.

Would you like more detailed guidance on any specific step, or do you have any other questions?

You

I now have this as my formula biut it didn't work:

```
Concatenate("Affected Departments: ",Concat( ForAll( ThisItem.'Request Departments', If( Value = "Other",
ThisItem.'Other Request Departments', Value ) ), ", "))
```

Copilot

I see where the issue might be. In Power Apps, the *Concat* function is used to concatenate strings within a table, while *Concatenate* is used to concatenate multiple strings. It looks like you're combining both functions, which might be causing the issue. Let's simplify it to use just the *Concat* function correctly:

Corrected Formula:

```
Concat("Affected Departments: ", Concat(
    ThisItem.'Request Departments',
    If(Value = "Other", ThisItem.'Other Request Departments', Value),
    ", "
))
```

However, since *Concat* inside another *Concat* is not directly supported, we can use an alternative approach by defining a variable or simplifying the concatenation. Here's the adjusted version:

1. **Define a Variable (Optional but Recommended for Clarity)**:
    - In the **OnSelect** property of a button or gallery item, set a variable to handle the concatenation.

```
Set(ConcatenatedDepartments, Concat(
    ForAll(
        ThisItem.'Request Departments',
        If(Value = "Other", ThisItem.'Other Request Departments', Value)
    ),
    ", "
));
```

2. **Use the Variable in the Text Property**:
    - In the label where you want to display the concatenated text, use:

```
Text: "Affected Departments: " & ConcatenatedDepartments
```

If you prefer not to use a variable, you can directly place the logic inside the label's **Text** property:

```
Text: "Affected Departments: " & Concat(
    ForAll(
        ThisItem.'Request Departments',
        If(Value = "Other", ThisItem.'Other Request Departments', Value)
    ),
    ", "
)
```

This should correctly concatenate all department values, replacing "Other" with the specified value, and add a comma between each entry.
Give this a try, and let me know if you encounter any issues or if there's anything else you need assistance with.

You

How can i take a text (*ThisItem.'Additional Info'*) and replace all the lien breaks with spaces?

Copilot

To replace all the line breaks with spaces in a text field (like *ThisItem.'Additional Info'*), you can use the Substitute function in Power Apps. Here's how you can achieve this:

Step-by-Step Guide:

1. **Use the Substitute Function**:
    - The *Substitute* function replaces occurrences of a specified substring with another string.
    - In this case, you will replace line breaks with spaces.
2. **Set the Text Property**:
    - Use the *Substitute* function in the *Text* property of the label to replace the line breaks.

Example:

Assume ThisItem.'Additional Info' contains the text with line breaks.

1. **Set the Text Property of a Label**:

```
Text: Substitute(ThisItem.'Additional Info', Char(10), " ")
```

Explanation:

- *Substitute(ThisItem.'Additional Info', Char(10), " ")*: Replaces all line breaks (represented by *Char(10)*) in the text with spaces.

Detailed Example:

1. **Select the Label**:
    - Click on the label where you want to display the text without line breaks.
2. **Set the Text Property**:
    - Enter the following formula:

```
Text: Substitute(ThisItem.'Additional Info', Char(10), " ")
```

This will replace all line breaks in the Additional Info field with spaces, making the text appear as a single line without breaks.
Would you like more detailed guidance on any specific step, or do you have any other questions?

You

One last thing, how can I refresh the gallery for when new data values are added or updated in the sharepoint list?

Copilot

To refresh the gallery in Power Apps when new data is added or updated in the SharePoint list, you can use the **Refresh** function. This function will reload the data from the data source, ensuring that the gallery displays the most current information. Here's how you can do it:

Step-by-Step Guide:

1. **Refresh the Data Source**:
   - Use the **Refresh** function to refresh the data source connected to your gallery.
2. **Update the Gallery**:
   - Ensure that the gallery's **Items** property is set to the data source so that it reflects the updated data.

Example:

1. **Add a Refresh Button**:
   - Insert a button in your app and set its **Text** property to "Refresh" or an icon of your choice.
2. **Set the OnSelect Property of the Button**:
   - Use the **OnSelect** property to refresh the data source and reset the gallery.
     OnSelect: Refresh(YourSharePointList); Reset(Gallery1)
3. **Automatically Refresh on Data Change**:
   - If you want the gallery to refresh automatically when data changes, you can set the **OnVisible** property of the screen to refresh the data source.
     OnVisible: Refresh(YourSharePointList)

Detailed Example:

1. **Button for Manual Refresh**:
   - Insert a button control.
   - Set its **Text** property to "Refresh".
   - Set its **OnSelect** property to:
     Refresh(YourSharePointList); Reset(Gallery1)
2. **Automatic Refresh on Screen Visible**:
   - Select the screen that contains the gallery.
   - Set the screen's **OnVisible** property to:
     OnVisible: Refresh(YourSharePointList)

Explanation:

- **Refresh(YourSharePointList)**: Refreshes the data source (your SharePoint list) to ensure it has the latest data.
- **Reset(Gallery1)**: Resets the gallery to reflect the refreshed data source.

This approach ensures that your gallery is always up-to-date with the latest information from your SharePoint list.
If you have any further questions or need more specific guidance, feel free to ask!

You

(Now Monday, Nov 18)
In a power automate flow, I have the following as my Input for a Compose connector, howver I'm getting a syntax error:

Input: concat('SAPCR#', padLeft(string(outputs('Compose')), 6, '0'))
Error: Unable to process template language expressions in action 'Compose_3' inputs at line '0' and column '0': 'The template function 'padLeft' is not defined or not valid.'.

Copilot

It looks like the *padLeft* function is not recognized in Power Automate. To achieve the same result, you can use a combination of other functions to pad the number with leading zeros. Here’s how you can do it:

Step-by-Step Solution:

1. **Initialize Variable**:
   - Initialize a variable to store the numeric part of the Request ID.
     Action: Initialize variable
     Name: NumericRequestID
     Type: String

2. **Compose Action to Increment the Value**:
   - Use the **Compose** action to increment the numeric part of the Request ID.
     Action: Compose
     Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)
3. **Compose Action to Pad the Number with Leading Zeros**:
   - Use the **Concat** and **SubString** functions to pad the number with leading zeros.
     Action: Compose
     Inputs: concat('SAPCR#', substring('000000', 0, 6 - length(string(outputs('Compose')))), string(outputs('Compose')))

Example Flow:

1. **Initialize Variable**:
   Action: Initialize variable
   Name: NumericRequestID
   Type: String
2. **Get Items from SharePoint**:
   - Ensure you have a step to get items and sort them by RequestID in descending order.
3. **Compose Action to Increment**:
   Action: Compose
   Inputs: add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)
4. **Compose Action to Pad the Number**:
   Action: Compose
   Inputs: concat('SAPCR#', substring('000000', 0, 6 - length(string(outputs('Compose')))), string(outputs('Compose')))

Explanation:

- *add(int(substring(last(body('Get_items'))?['RequestID'], 6, 6)), 1)*: Increments the numeric part of the Request ID.
- *concat('SAPCR#', substring('000000', 0, 6 - length(string(outputs('Compose')))), string(outputs('Compose')))*: Pads the number with leading zeros and concatenates it with "SAPCR#".

This approach should resolve the syntax error and correctly format the new Request ID.
Give this a try, and let me know if you encounter any issues or if there's anything else you need assistance with.

You

I have the following Python code, how can I write it in a compose connector in the Power automate flow:

```
if oldRequestID.startsWith("SAPCR#"):
    return int(oldRequestID[10: ])
else:
    return 100
```

Copilot

You can implement the logic from your Python code in a Compose action within a Power Automate flow using expressions. Here's how you can translate your Python code to Power Automate expressions:

Python Code:

```
if oldRequestID.startsWith("SAPCR#"):
    return int(oldRequestID[10:])
else:
    return 100
```

Power Automate Equivalent:

1. **Compose Action**:
   - Use a Compose action to replicate the if-else logic.
2. **Expression**:
   - Use the *if* function to check the condition.
   - Use the *substring* function to extract the part of the string.
   - Use the *int* function to convert the string to an integer.

Example Compose Action:

```
if(startsWith(outputs('ComposeName')['oldRequestID'], 'SAPCR#'), int(substring(outputs('ComposeName')['oldRequestID'], 6, 6)), 100)
```

Explanation:

- startsWith(outputs('ComposeName')['oldRequestID'], 'SAPCR#'): Checks if oldRequestID starts with "SAPCR#".
- *substring(outputs('ComposeName')['oldRequestID'], 6, 6)*: Extracts the numeric part starting from the 6th character. Adjust the length (last parameter) if the actual length is different.

- *int(...)*: Converts the extracted substring to an integer.
- *if(..., ..., 100)*: If the condition is true, it returns the converted integer; otherwise, it returns 100.

Step-by-Step Guide:

1. **Add a Compose Action**:
   - In your flow, add a Compose action where you want to implement this logic.
2. **Set the Compose Input**:
   - Set the input to the following expression:

     ```
     if(startsWith(outputs('ComposeName')['oldRequestID'], 'SAPCR#'),
     int(substring(outputs('ComposeName')['oldRequestID'], 6, 6)), 100)
     ```
   - Replace *ComposeName* with the actual name of the step that contains the *oldRequestID*.

This expression will replicate the logic of your Python code, checking if *oldRequestID* starts with "SAPCR#" and returning the appropriate integer value.
Would you like any further assistance or details on this?

Conversation by Fifa Utuber with Copilot on Wednesday, Nov 6 2024