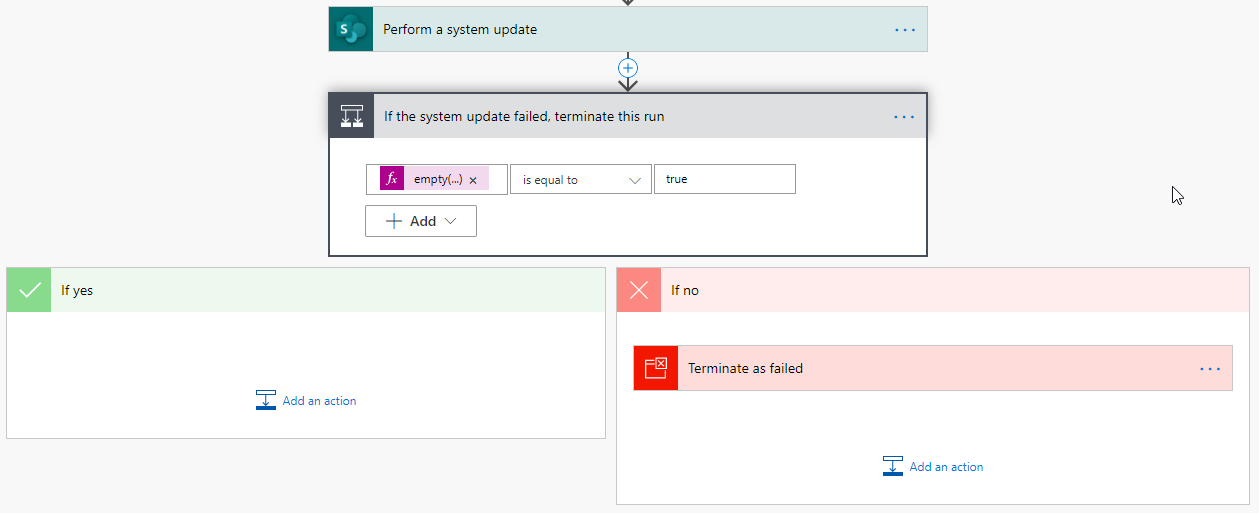
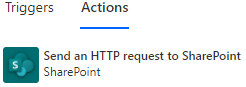
Performing a SharePoint SystemUpdate call in Power Automate to Avoid Infinite Loops

\More details are here: [smayes3/azure-logic-apps-sharepoint-systemupdate: A generic solution to call the SharePoint SystemUpdate() method in Azure Logic Apps and Microsoft Power Automate (dxc.com)](https://github.dxc.com/smayes3/azure-logic-apps-sharepoint-systemupdate)

When ready to update data in the same SharePoint list item that triggered the current flow, instead of using the **Update item** action in the **SharePoint** connector which will trigger an infinite loop, use the following three actions and configure them as follows. These actions simulate a CSOM request using the SystemUpdate() method on the ListItem object to cause SharePoint to update field values without triggering an updated **Modified By** user, an updated **Modified** time stamp, and an update event.



# Perform a system update action

This utilizes the **Send an HTTP request to SharePoint** action in the **SharePoint** connector. Start by selecting that action.

It’s suggested to rename this action to **Perform a system update** or some other memorable name. This name will be used in later formulas, so give it a name now so you do not have to worry about adjusting other formulas later.

Populate the following properties:

* **Site Address** *Use the same site as the trigger*
* **Method** POST
* **Uri** \_vti\_bin/client.svc/ProcessQuery
* **Headers** Accept \*/\*  
   Content-Type text/xml;charset="UTF-8"

Next, construct the body of the request. Start with the XML below. This XML can be pasted directly into the Power Automate action.

<Request xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" SchemaVersion="15.0.0.0" LibraryVersion="16.0.0.0" ApplicationName="Power Automate">

<Actions>

METHODS

</Actions>

<ObjectPaths>

<StaticProperty Id="0" TypeId="{3747adcd-a3c3-41b9-bfab-4a64dd2f1e0a}" Name="Current" />

<Property Id="1" ParentId="0" Name="Web" />

<Method Id="2" ParentId="1" Name="GetListItem">

<Parameters>

<Parameter Type="String">SERVER\_RELATIVE\_WEB\_URL/@{triggerBody()?['{FullPath}']}</Parameter>

</Parameters>

</Method>

</ObjectPaths>

</Request>

For the two highlighted sections above, we’ll replace those areas with content relative to the updates to be made.

For SERVER\_RELATIVE\_WEB\_URL, replace it with the server-relative URL of the trigger’s site. For example, if the trigger is set to use the site address **https://smayeshp.sharepoint.com/sites/ModernTeam**, then the server-relative URL is **/sites/ModernTeam**

For METHODS, create a section of XML for each field value that needs to be updated. These sections represent setting each field value and can be repeated as many times as necessary; once for each field value that needs to be updated. Lastly, we’ll have one final section that will represent the call to the SystemUpdate() method.

For each field value that needs to be updated, use the following XML.

<Method Name="SetFieldValue" Id="METHOD\_ID" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[FIELD\_INTERNAL\_NAME]]></Parameter>

<Parameter Type="VALUE\_TYPE"><![CDATA[UPDATED\_VALUE]]></Parameter>

</Parameters>

</Method>

For METHOD\_ID, start with the number 3 and increment this value for each field. For example, if updating the values for three fields, the first field would use the number 3, the second field would use 4, and the third field would use 5.

For FIELD\_INTERNAL\_NAME, enter the exact internal name of the field that will be updated.

For VALUE\_TYPE, enter the data type for corresponding to the field value: **String** for text, **Integer** for numbers, etc.

For UPDATED\_VALUE, enter the value; which may likely be a reference to a variable or a custom expression.

Putting together a full example of the METHODS XML, consider updating a **Status** field to **In Progress** and a **Completed** field to **No**. The XML would be built as follows.

<Method Name="SetFieldValue" Id="3" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[Status]]></Parameter>

<Parameter Type="String"><![CDATA[In Progress]]></Parameter>

</Parameters>

</Method>

<Method Name="SetFieldValue" Id="4" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[Completed]]></Parameter>

<Parameter Type="String"><![CDATA[No]]></Parameter>

</Parameters>

</Method>

The final section of XML we need to add to the METHODS section instructs the API to call the SystemUpdate() method. Use the following XML.

<Method Name="SystemUpdate" Id="SYSTEM\_UPDATE\_METHOD\_ID" ObjectPathId="2" />

For SYSTEM\_UPDATE\_METHOD\_ID, use the number following the last METHOD\_ID number used above it. In the example above, if 4 was the last METHOD\_ID number used, then enter 5 for SYSTEM\_UPDATE\_METHOD\_ID.

Given the above examples, our full body for this request would be as follows.

<Request xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" SchemaVersion="15.0.0.0" LibraryVersion="16.0.0.0" ApplicationName="Power Automate">

<Actions>

<Method Name="SetFieldValue" Id="3" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[Status]]></Parameter>

<Parameter Type="String"><![CDATA[In Progress]]></Parameter>

</Parameters>

</Method>

<Method Name="SetFieldValue" Id="4" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[Completed]]></Parameter>

<Parameter Type="String"><![CDATA[No]]></Parameter>

</Parameters>

</Method>

<Method Name="SystemUpdate" Id="5" ObjectPathId="2" />

</Actions>

<ObjectPaths>

<StaticProperty Id="0" TypeId="{3747adcd-a3c3-41b9-bfab-4a64dd2f1e0a}" Name="Current" />

<Property Id="1" ParentId="0" Name="Web" />

<Method Id="2" ParentId="1" Name="GetListItem">

<Parameters>

<Parameter Type="String">/sites/ModernTeam/@{triggerBody()?['{FullPath}']}</Parameter>

</Parameters>

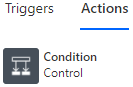
</Method>

</ObjectPaths>

</Request>

# If the system update failed, terminate this run

Unless there is an exceptional problem, the CSOM call will return an HTTP 200 response even if there is an error. If an error occurred, it would be in the body of the response. It is the flow author’s responsibility to review the response from the previous call and terminate the flow with a failed status if an error occurred.

This action utilizes the **Condition** action in the **Control** connector. Start by selecting that action.

It’s suggested to rename this action to **If the system update failed, terminate this run** or some other memorable name.

Set the condition to the following:

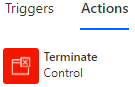
empty(body('Perform\_a\_system\_update')?[0]?['ErrorInfo'])

is equal to

true

Note: Should the name of the first action been set to something other than **Perform a system update**, then be sure to update the first part of the condition. Note that any space in the name is replaced with an underscore in the formula.

# Terminate as failed

 This action utilizes the **Terminate** action in the **Control** connector. Start by selecting that action within the **If no** side of the condition that was just created in the prior step.

Populate the following properties:

* **Status** Failed
* **Code** @{body('Perform\_a\_system\_update')?[0]?['ErrorInfo']?['ErrorCode']}
* **Message**@{body('Perform\_a\_system\_update')?[0]?['ErrorInfo']?['ErrorMessage']}  
    
  Error Type Name: @{body('Perform\_a\_system\_update')?[0]?['ErrorInfo']?['ErrorTypeName']}  
  Trace Correlation ID: @{body('Perform\_a\_system\_update')?[0]?['ErrorInfo']?['TraceCorrelationId']}

Note: Should the name of the first action been set to something other than **Perform a system update**, then be sure to update the first part of the condition. Note that any space in the name is replaced with an underscore in the formula.

# Congratulations! You’re now calling the SystemUpdate() CSOM method from a Flow in Power Automate!

# Field Types other than string:

[modernize-sharepoint-applications/setfieldvalue-field-value-parameter.md at main · SharePointMigrations/modernize-sharepoint-applications (dxc.com)](https://github.dxc.com/SharePointMigrations/modernize-sharepoint-applications/blob/main/azure-logic-apps-power-automate/system-update/setfieldvalue-field-value-parameter.md)

Note: for a hyperlink, the 2nd Parameter type = “String” and the Value is URL, Description. Note that there is a space between the comma and the Description value.

Example:

<Method Name="SetFieldValue" Id="3" ObjectPathId="2">

<Parameters>

<Parameter Type="String"><![CDATA[WebLink]]></Parameter>

<Parameter Type="String"><![CDATA[https://www.google.com, Google]]></Parameter>

</Parameters>

You can use Dynamic content for the URL and/or the Description values.