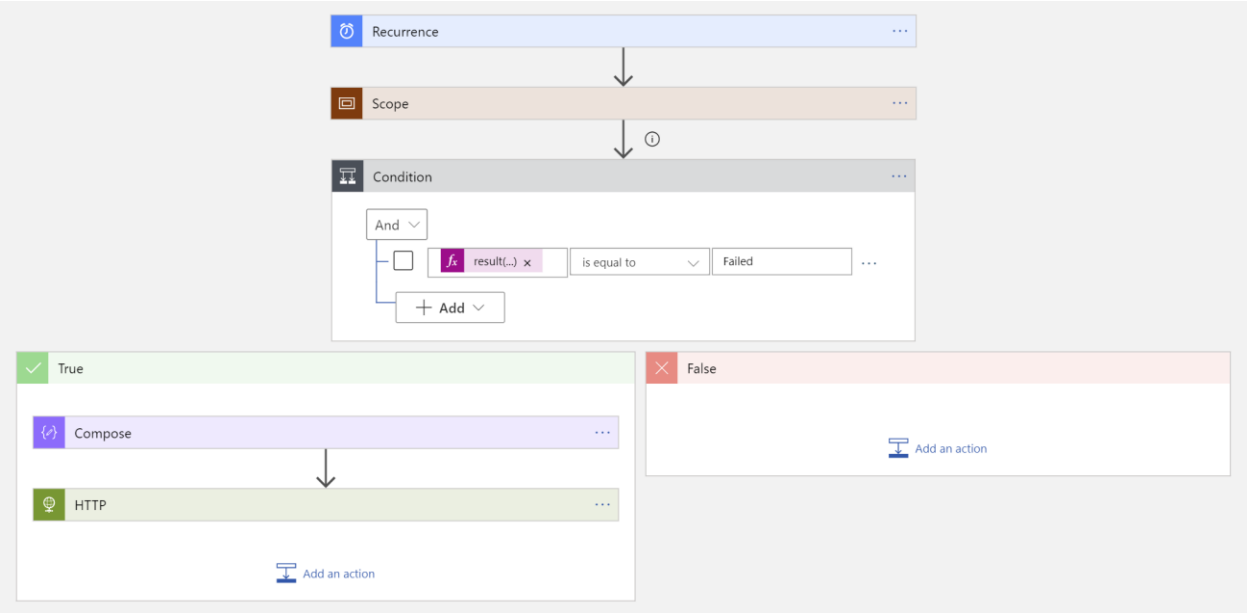
Logic Apps | Resubmit RunId automatically when an action fails.

The solution described in this document is to resubmit a workflow run when a specific action fails. This is done using Rest API.

Caution: If the action in scope is failing without any way to automatically succeed while resubmitting the workflow run, this can create an endless loop where the Logic App workflow will be executed over and over again and would only stop if the action in scope succeeds, the Logic App workflow is cancelled before automatic resubmission occurs, or the Logic App is disabled.

Design overview



Steps to follow and related documentation

Add A trigger

In this example we use a recurrence trigger

https://docs.microsoft.com/en-us/azure/logic-apps/tutorial-build-schedule-recurring-logic-app-workflow#add-the-recurrence-trigger

Add a Scope

The next step is to add a scope, this will be used to capture the error of the action added within the scope

https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-control-flow-run-steps-group-scopes#add-steps-to-scope

Add an Action in the scope

Add the action you want the workflow automatically checks if it fails to resubmit the workflow run.

Add a Condition step

After the Scope you can add a condition.

https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-control-flow-conditional-statement

The condition can be the following:

result('Scope')[0]['status']	is equal to	Failed

Or the condition can check status code. In the example below the workflow will be resubmitted only when the action in question fails with code 401

result('Scope')[0]?['outputs']?['statuscode']	is equal to	401

If the action within the scope fails, the result will be True which means the workflow will go to the true section where it resubmits the workflow run. If the action within the scope succeeds, the result will be false which means the workflow will go to the False section where it does not resubmit the workflow run.

Configure run after in the condition step

To allow the workflow run keeps running after the action fails, we need to configure the run after in the condition step.

https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-exception-handling#catch-and-handle-failures-by-changing-run-after-behavior

Call Rest API from Logic Apps

To call Rest API we can use HTTP action

https://docs.microsoft.com/en-us/azure/connectors/connectors-native-http#add-an-http-action

For this example, we will resubmit a RunId by using Rest API (Workflow Trigger Histories – Resubmit)

https://docs.microsoft.com/en-us/rest/api/logic/workflowtriggerhistories/resubmit

POST
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Logic/workflows/{workflowName}/triggers/{triggerName}/histories/{historyName}/resubmit?api-version=2016-06-01

To successfully execute the Rest API above there are two considerations we need to take care of:

- Workflow Trigger Histories – Resubmit URL needs a history name which is a RunId, so we first need to get the RunId, which is done by using a workflow function and compose action.

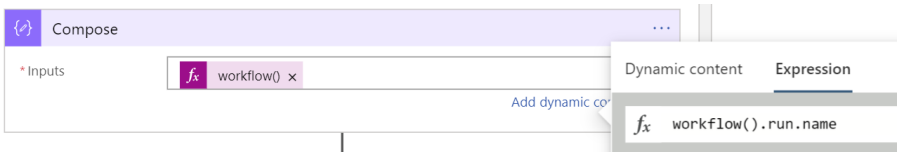
- Resubmit Rest API requires authentication, to allow this we can enable managed identity on the Logic App and assign it the Logic App contributor role to grant access to itself.
  https://docs.microsoft.com/en-us/azure/logic-apps/create-managed-service-identity
  https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles#logic-app-contributor

Get the RunId

We can use a compose action to capture the workflow's current run using a workflow function.

workflow().run.name

Example:

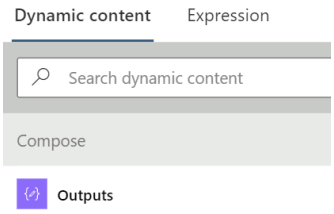


https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-perform-data-operations#compose-action

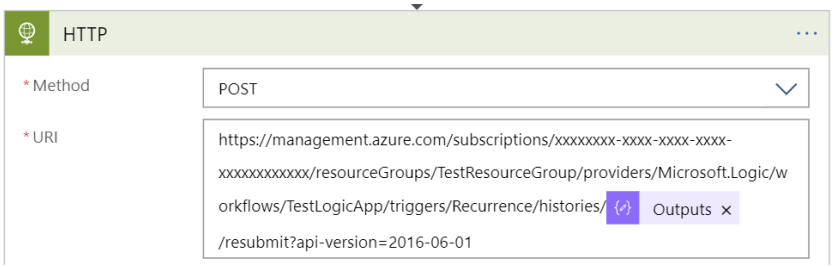
https://docs.microsoft.com/en-us/azure/logic-apps/workflow-definition-language-functions-reference#workflow

Add HTTP action to resubmit the RunId

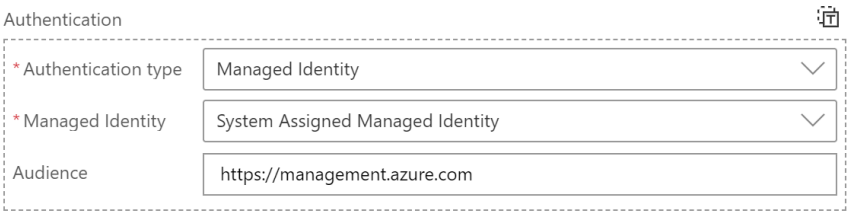
After adding the http action, select POST in the method field, then modify the sample URL in the REST API link below according to your Logic App resource ID and for the historyname you can select the "Outputs" token of the Compose output in the Dynamic content.



Example:



Then, set up the authentication parameter, example:



https://docs.microsoft.com/en-us/azure/connectors/connectors-native-http#add-an-http-action

https://docs.microsoft.com/en-us/rest/api/logic/workflowtriggerhistories/resubmit

After completing all the steps described you can test your Logic App.