**Implementing Try, Catch and Finally in Power Automate**

No matter how great we are in programming , sometimes our code has errors. They may occur because of our mistakes, an unexpected user input, an erroneous server response, and for a thousand other reasons.

So, handling these exceptions is very important . Anyone who has even a little experience with programming knows the **"try – catch"** to handle these exceptions.

Try, Catch, finally blocks in the above template are nothing but scope controls

**Scopes** are a simple way to group two or more actions together. When you put actions inside a scope, they can be visually collapsed so that you cake make the presentation inside of the designer much simpler.

**Try** – The “Try” scope should contain all the actions from the main flow of the process.

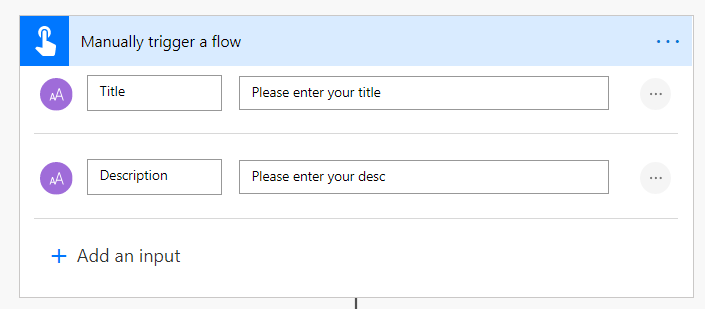
**Catch** – The “Catch” scope is configured to run after “Try” block is failed. This is implemented using “Configure run after “ feature .

**Finally** – The “Finally” block should run no matter what happens in the previous action.

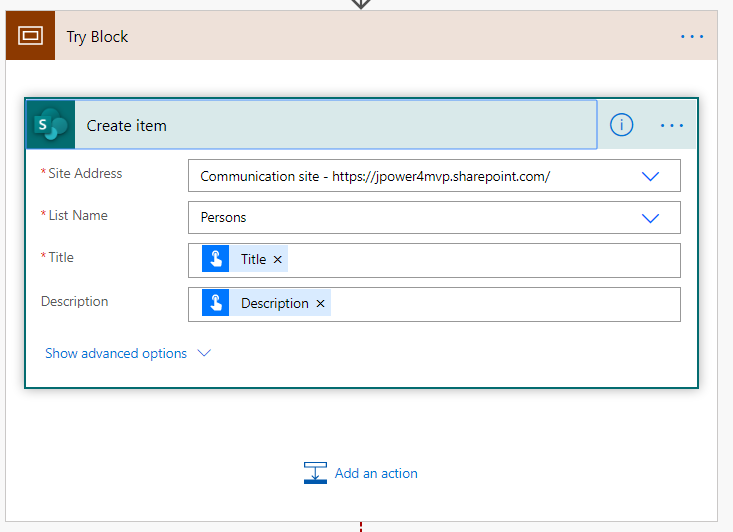
Using “**Scope**” and “**Configure run after**” feature, we can implement try ,catch pattern in Power Automate easily.

**Steps to implement try, catch and finally in Power Automate**

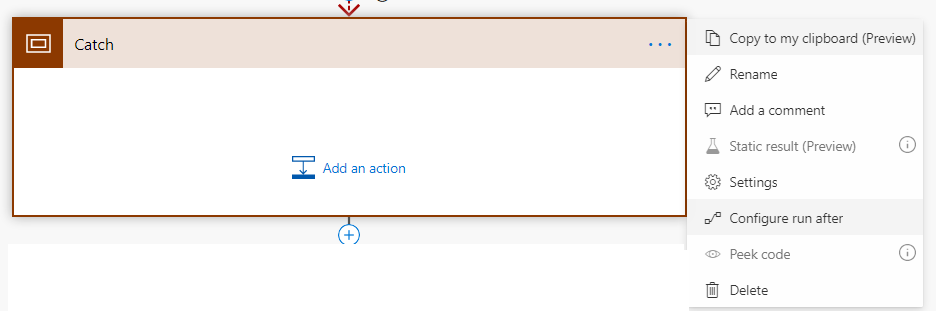
1. Login flow.microsoft.com
2. Create an Instant cloud flow
3. Add two input columns
   1. Title
   2. Description



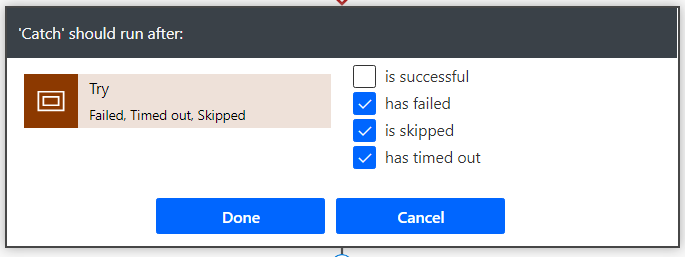
1. Add an action Scope to group Try block
2. Then add an action Create item to save the data to SharePoint



1. Add a new action Scope to group Catch block



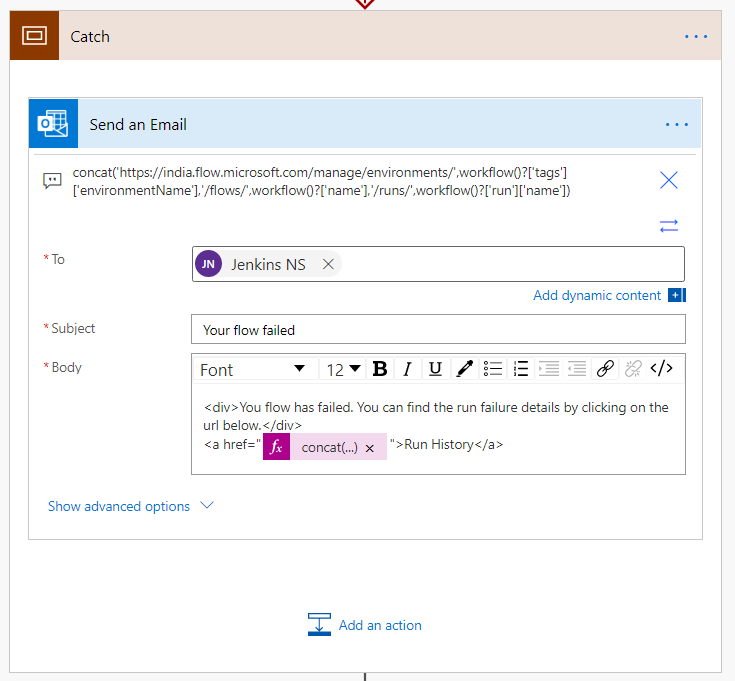
1. Using scope and configure run after feature to implement catch pattern



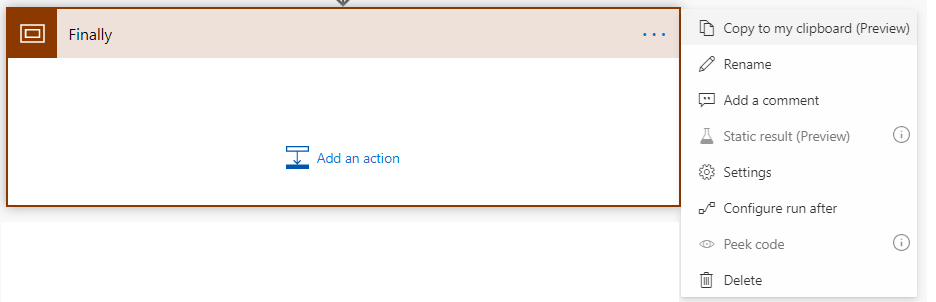
1. Click done and should run after flow has failed
2. Then add an action to send email to notify the user

*concat('https://india.flow.microsoft.com/manage/environments/',workflow()?['tags']['environmentName'],'/flows/',*

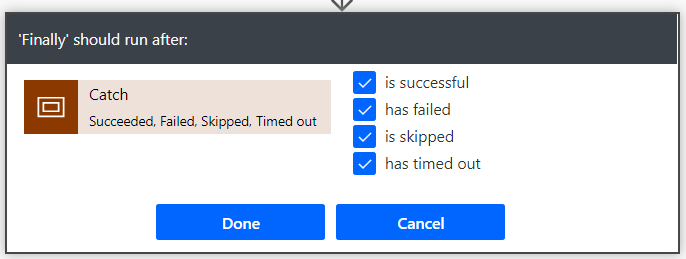
*workflow()?['name'],'/runs/',workflow()?['run']['name'])*



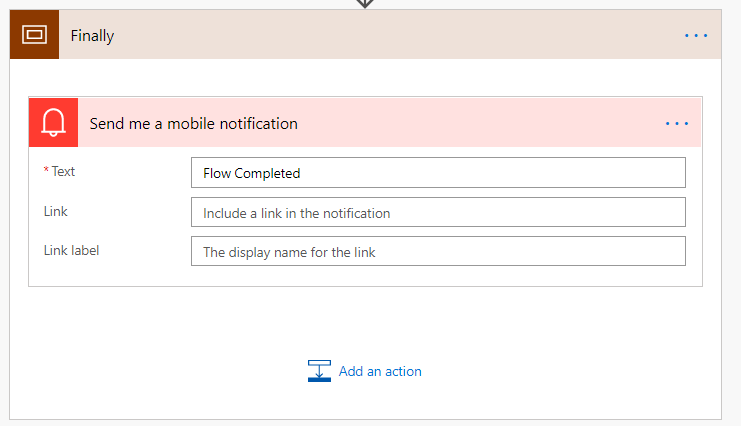
1. Then add scope action to manage finally block



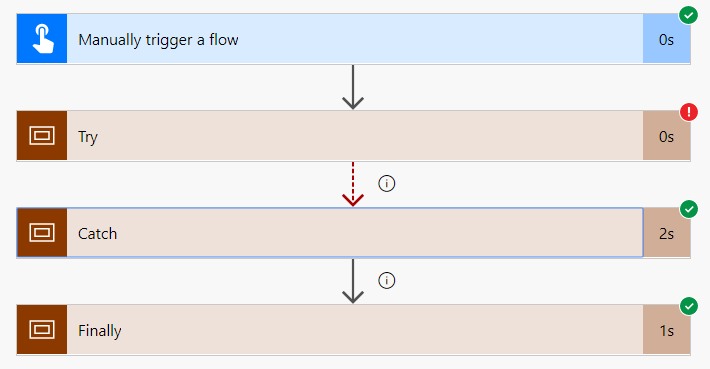
1. Using scope and configure run after feature to implement finally pattern



1. Click done to activate the block
2. Add action to send a push notification to the user



1. Save and test the flow
2. Output if I add wrong URL



Add body from SharePoint Cerate Item action output

