1. Create a new logic app in the portal
2. Add an http receive trigger.
3. Use the Orders.json in the folder as a template for the type of document that will be passed in.
4. Add a for-each condition
5. Add the Orders object for looping
6. Add an Azure function, the function that puts an item into a storage queue
7. The syntax should be {“name” : “CurrentItem”} Current item should be selected from the dynamic properties
8. Retrieve the http address from the function.
9. Open Postman
10. Setup Postman to do a Post to the http address
11. Add an http header with a content-type of application/json
12. Click on Body and then raw

Paste in the value of Orders.json  
  
{

"Orders": [

{

"orderId": 1,

"customerName": "Customer 1",

"productId": 1000,

"itemCount": 1,

"itemPrice": 1.55

},

{

"orderId": 2,

"customerName": "Customer 2",

"productId": 1001,

"itemCount": 1,

"itemPrice": 12.95

}

]

}

1. Post the command in Postman and then look in the storage queue
2. Add a new Scope shape to the workflow and then move it up above the looping control
3. Drag the loop control into the scope, rename the scope to DoWork
4. Create another scope and call it ErrorHandler.
5. Click on the scope properties and then set the RunAfter to failed.
6. Create another scope and call it Success
7. Click on the scope properties and then set the RunAfter to Skipped
8. In the ErrorHandler scope, add a SendEmailV2 action.
9. Body: fx: Outputs, Subject: Failed queue operation To: yourself
10. Go to the Success Scope, add an email action, same information as before except subject should be Success (of some sort)
11. Go back into the work scope and change the HttpToAzureQueue request to something other than name
12. Do another post from postman, see your email with errors
13. Change the request body to the HttpToAzureQueue back to ‘name’ and do another post. Wait for the success email.