Contents

[Introduction 1](#_Toc44254350)

[Flow Design 1](#_Toc44254351)

[Dataweave Transformation 1](#_Toc44254352)

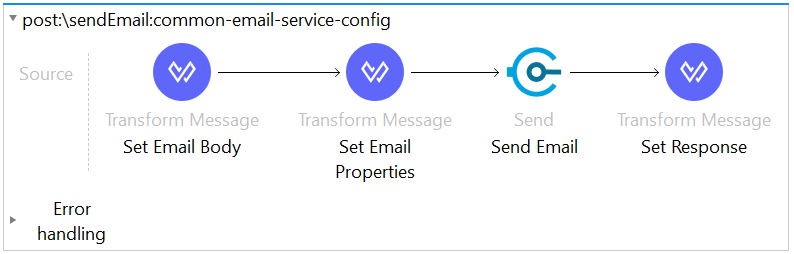
[Test Cases 4](#_Toc44254353)

# Introduction

Common Email Service can accept XML/JSON and convert it to HTML with two-column HTML-Table.

Even the nested-(key-value) pairs will be mapped to HTML table. Please refer [Test-Cases](#_Test_Cases) section for more details.

# Flow Design



# Dataweave Transformation

* **Generate HTML Body from Input Payload (Set Email Body in above Flow Design )**

|  |
| --- |
| %dw 2.0  import \* from dw::core::Strings  import \* from dw::Core  output application/xml  var cssStyle={  "table": {  border: 3,  cellpadding: 2,  cellspacing: 1,  style: "width:100%"  },  "tableHeader": {  style: "text-align: center;",  bgcolor: "yellow",  colspan: "2"  },  "tableBody" : {  style: "text-align: left;"  }  }  var noMapKeyList=["emailDetails"]  var tableHeader={      "true": "Error Report",      "false": "Notification"  }  var mailHeader={      "true": ({  p: "Hi,",                  p: "Please find below Error Report. Take Appropriate Actions if Applicable!"              }),      "false":({  p: "Hi,",                  p: "Please find below Notification on recent request!"  })  }  var mailFooter={  p: "Note: This is a System generated e-mail, Please do not reply."  }  fun nullEmptyCheck(value)=(  (value != null and value != "")  )  fun checkNoMapKeyList(key)=(  !(noMapKeyList contains key as String)  )  fun checkIfObject(value)=(  (value is Object)  )  fun generateTableHeader()=(  tr: {  th @((cssStyle["tableHeader"])): tableHeader[nullEmptyCheck(payload.errorDetails)]  }  )  fun generateTableRow(key, value)=(  (tr: {  td : (capitalize(key as String)),  td : value  })if(checkNoMapKeyList(key) and !(value is Array))  )  fun mapInputPayload (payload)=(  (payload mapObject (value, key, index) ->  if(checkNoMapKeyList(key) and checkIfObject(value))  (mapInputPayload(value))  else (  generateTableRow(key, value)  )  )  )  ---  {      html: {          body: {              header: (mailHeader[nullEmptyCheck(payload.errorDetails)]),              table @((cssStyle["table"])): {                  thead: generateTableHeader(),                  tbody @((cssStyle["tableBody"])): mapInputPayload(payload)              },              footer: mailFooter          }      }  } |

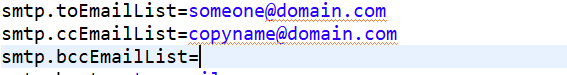
* **Email Properties to capture distribution List (Set Email Properties in above Flow Design)**

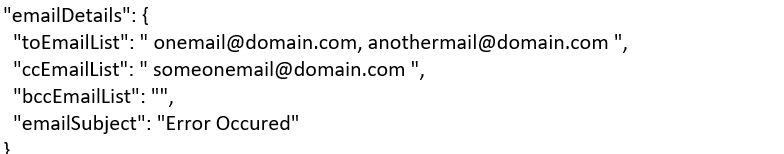
|  |
| --- |
| %dw 2.0  **output** **application/java**  **fun** nullEmptyCheck(value)=(  (value != **null** **and** value != "")  )  ---  {  toList: **if**(nullEmptyCheck(payload.emailDetails.toEmailList)) (payload.emailDetails.toEmailList splitBy ",") **else** (p('smtp.toEmailList') splitBy ","),  ccList: **if**(nullEmptyCheck(payload.emailDetails.ccEmailList)) (payload.emailDetails.ccEmailList splitBy ",") **else** (p('smtp.ccEmailList') splitBy ","),  bccList: **if**(nullEmptyCheck(payload.emailDetails.bccEmailList)) (payload.emailDetails.bccEmailList splitBy ",") **else** (**if** (nullEmptyCheck(p('smtp.bccEmailList'))) p('smtp.bccEmailList') splitBy "," **else** **null**),  emailSubject: **if**(payload.errorDetails != **null**) "Error Report" **else** "Notification"  } |

# Test Cases

1. Error Notification With Email Details Object

|  |
| --- |
| {  "environment": "Dev",  "correlationId": "20200622T130642449",  "instanceId": "IT130642449Z",  "businessProcessName": "Elgibility Check",  "emailDetails": {  "toEmailList": " onemail@domain.com, anothermail@domain.com ",  "ccEmailList": " someonemail@domain.com ",  "bccEmailList": "",  "emailSubject": "Error Occured"  },  "errorDetails": {  "isRetryable": false,  "technicalErrorType": "APIKIT:BAD\_REQUEST",  "errorType": "BUSINESS\_ERROR",  "applicationErrorCode": "10123",  "errorSeverity": "High",  "errorDescription": "Missing Query Parameter testme"  }  } |
| Mail Sent TO: **onemail@domain.com, anothermail@domain.com**  Mail Sent CC: **someonemail@domain.com**  Mail Sent BCC: **None** |

If you want to override distribution list mentioned in **common-email-service** API.  


You can mention in **emailDetails** like below  


1. Error Notification Without emailDetails:

|  |
| --- |
| {  "environment": "Dev",  "correlationId": "20200622T130642449",  "instanceId": "IT130642449Z",  "businessProcessName": "Elgibility Check",  "errorDetails": {  "technicalErrorType": "APIKIT:BAD\_REQUEST",  "errorType": "BUSINESS\_ERROR",  "applicationErrorCode": "10123",  "errorSeverity": "High",  "errorDescription": "Missing Query Parameter testme"  }  } |
| **Email Distribution List Configured in Common Email Service API.**    Mail Sent TO: **someone@domain.com**  Mail Sent CC: **copyname@domain.com**  Mail Sent BCC: **None** |

1. Generic Notification

|  |
| --- |
| {  "correlationId": "20200622T130642449",  "instanceId": "IT130642449Z",  "appName": "eligibility-api",  "flowName": "process-eligibility-request-flow",  "businessProcessName": "Elgibility Check",  "notificationDetails": {  "status": "Status/Initiated/Completed",  "message": "Request has been Submitted",  "action": "Check the Status on Portal"  }  } |
| **Email Distribution List Configured in Common Email Service API.**    Mail Sent TO: **someone@domain.com**  Mail Sent CC: **copyname@domain.com**  Mail Sent BCC: **None** |

1. Generic Notification With Dynamic Key-Value Pairs

|  |
| --- |
| {  "correlationId": "20200622T130642449",  "instanceId": "IT130642449Z",  "appName": "eligibility-api",  "flowName": "process-eligibility-request-flow",  "businessProcessName": "Elgibility Check",  "notificationDetails": {  "additionalDetails": "Request has been Initiated",  "status": "Initiated",  "message": "Request has been Submitted",  "action": "Check the Status on Portal"  }  } |
| **Email Distribution List Configured in Common Email Service API.**    Mail Sent TO: **someone@domain.com**  Mail Sent CC: **copyname@domain.com**  Mail Sent BCC: **None** |

1. Generic Notification with Email Details in Request Body

|  |
| --- |
| {  "correlationId": "20200622T130642449",  "instanceId": "IT130642449Z",  "appName": "eligibility-api",  "flowName": "process-eligibility-request-flow",  "businessProcessName": "Elgibility Check",  "emailDetails": {  "toEmailList": " onemail@domain.com, anothermail@domain.com ",  "ccEmailList": " someonemail@domain.com ",  "bccEmailList": "",  "emailSubject": "Error Occured"  },  "notificationDetails": {  "status": "Completed",  "message": "Request has been Closed as the Necessary Actions are Fulfilled"  }  } |
| Since the Email Distribution list is passed in Input Payload.  Email distribution lists in Properties files are ignored.  Mail Sent TO: **onemail@domain.com, anothermail@domain.com**  Mail Sent CC: **someonemail@domain.com**  Mail Sent BCC: **None** |