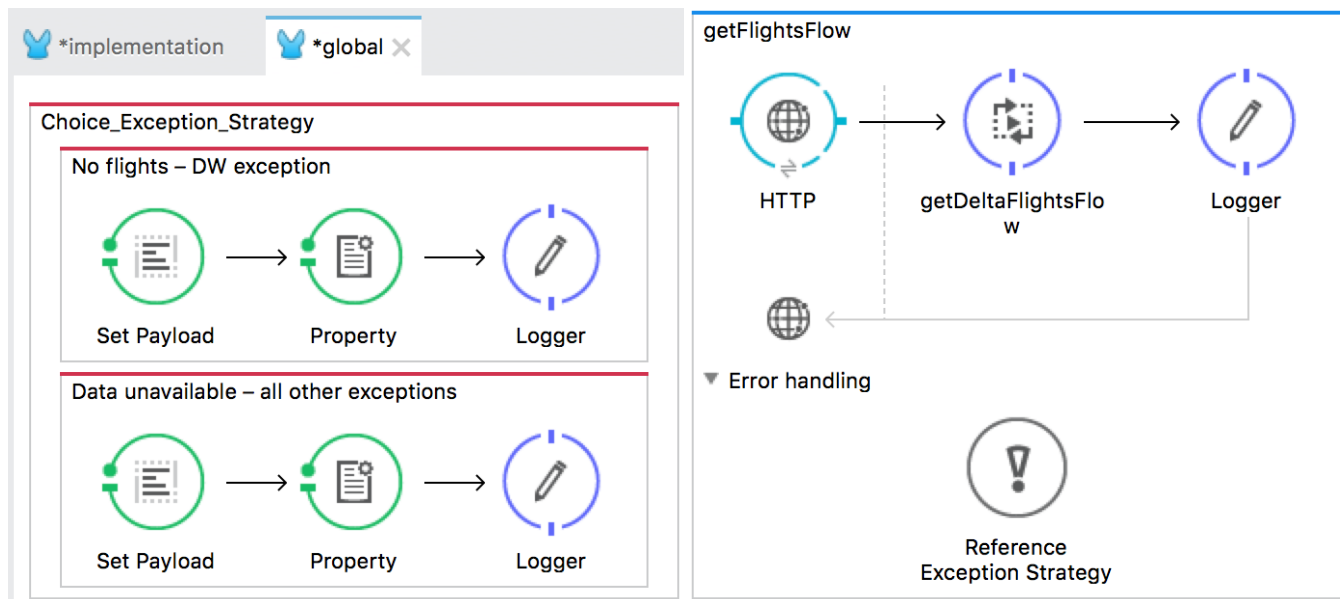


# Module 9: Handling Errors



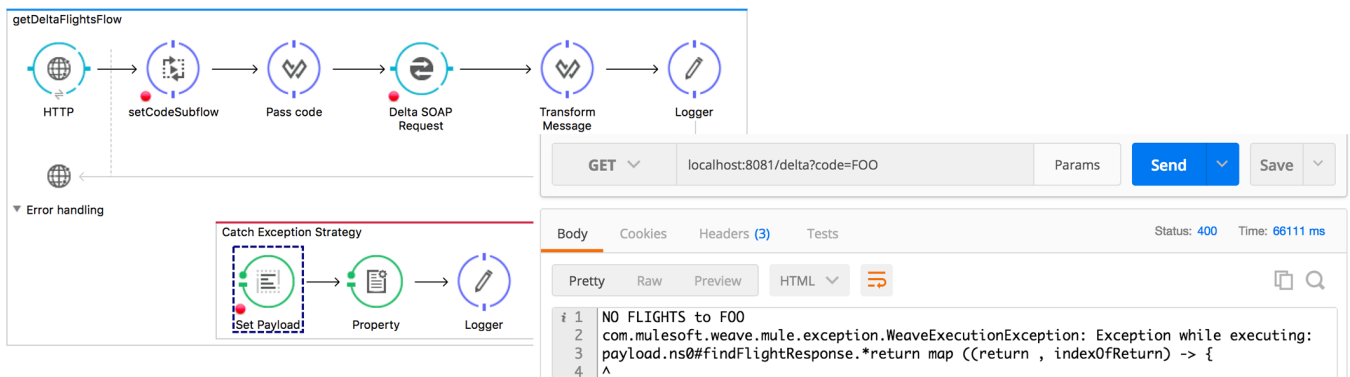
**At the end of this module, you should be able to:**

- Describe the different types of exception strategies.
- Handle messaging exceptions in flows.
- Create and use global exception handlers.
- Specify a global default exception strategy.

## Walkthrough 9-1: Handle a messaging exception

In this walkthrough, you handle an exception thrown by the Delta flow when a destination with no flights is used. You will:

- Add a catch exception strategy to a flow.
- Catch an exception and set the payload to send an error message back.
- Reference an exception object inside an exception handler.
- Set an HTTP status code inside an exception handler.



## Debug the application for a request with a non-existent destination

1. Return to getDeltaFlightsFlow.
2. Make sure there is a breakpoint on the flow reference component.
3. Debug the project.
4. In Postman, make another request to <http://localhost:8081/delta?code=FOO>.
5. In the Mule Debugger, step through the application and when you get the exception, drill-down into the exceptionThrown object.

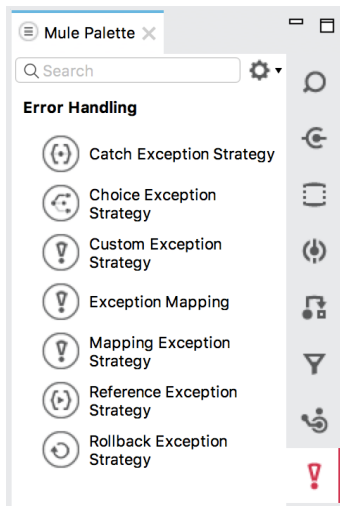
The screenshot shows the Mule Debugger interface with the 'Mule Properties' tab selected. The 'exceptionThrown' object is expanded, showing its properties and values. The 'cause' property is highlighted, showing the exception message: 'com.mulesoft.weave.mule.exception.WeaveExecutionException: Exception while executing: payload.ns0#findFlightResponse.\*return map ((return , indexOfReturn) -> {'. The 'errorCode' property is -1, and the 'event' property is a MuleEvent object.

Name	Value	Type
DataType	SimpleDataType{type=org.apache.cxf.staxuti...	org.mule.transformer.types.SimpleDataType
Exception	null	
exceptionThrown	com.mulesoft.weave.mule.exception.WeaveE...	com.mulesoft.weave.mule.exception.WeaveE...
cause	com.mulesoft.weave.engine.ast.dynamic.Une...	com.mulesoft.weave.engine.ast.dynamic.Une...
CAUSE_CAPTION	Caused by:	java.lang.String
causeRollback	false	java.lang.Boolean
detailMessage	com.mulesoft.weave.engine.ast.dynamic.Une...	java.lang.String
EMPTY_THROWABLE_ARRAY	[Ljava.lang.Throwable;@1fc5724b	java.lang.Throwable[]
errorCode	-1	java.lang.Integer
event	MuleEvent: 0-28442770-1c52-11e6-b0a8-...	org.mule.DefaultMuleEvent
EXCEPTION_MESSAGE_DELIMITER	*****	java.lang.String
EXCEPTION_MESSAGE_SECTION_...	-----	java.lang.String
failingMessageProcessor	null	org.mule.api.processor.MessageProcessor
handled	false	java.lang.Boolean

- Click the Resume button.

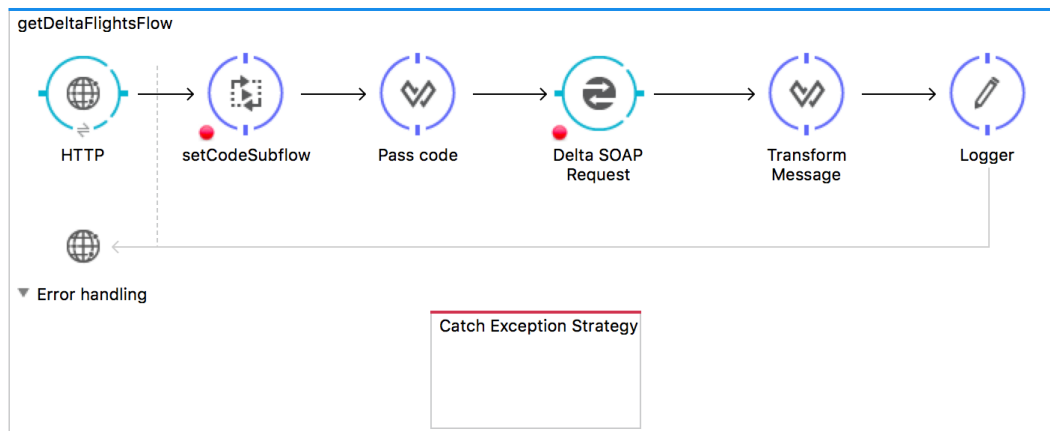
## Browse the error handling elements in the Mule Palette

- Switch perspectives and in the Mule Palette, select the Error Handling tab.
- View the available error handling processors.



## Add a catch exception strategy

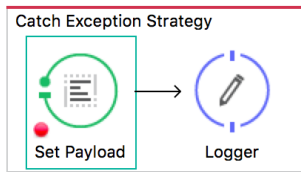
- In getDeltaFlightsFlow, click the arrow to expand the Error handling section.
- Drag a Catch Exception Strategy from the Mule Palette into the error handling section of the flow.



- Add a Set Payload transformer to the catch exception strategy.
- In the Set Payload properties view, set the value the following MEL expression:

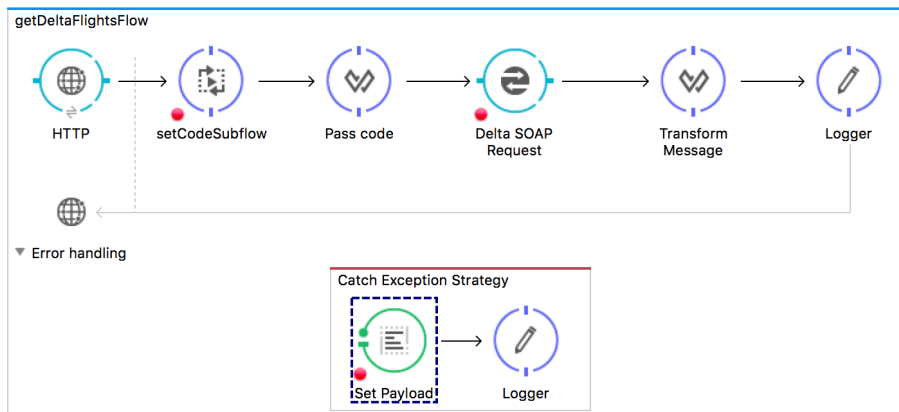
NO FLIGHTS to #[flowVars.code + '\n' + exception]

13. Make sure there is a breakpoint on the transformer inside the catch exception.
14. Add a Logger after the transformer.

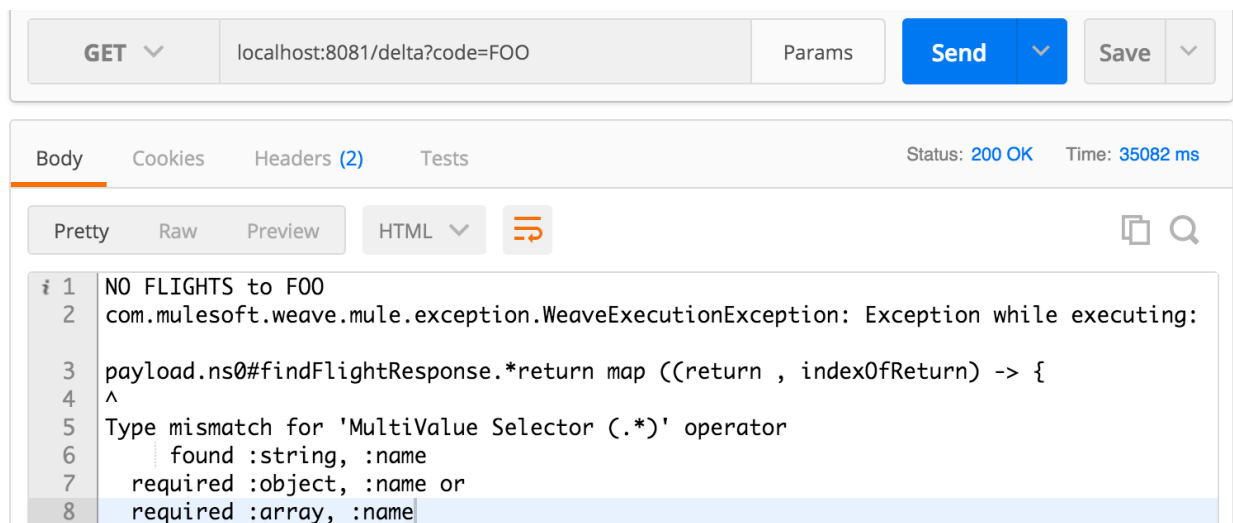


## Test the application

15. Redeploy the application in debug mode.
16. In Postman, make another request to <http://localhost:8081/delta?code=FOO>.
17. Step through the application; you should see the exception thrown in getDeltaFlightsFlow and handled by the exception handler.

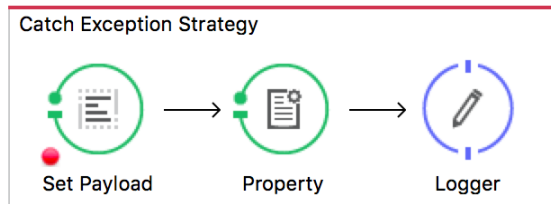


18. Step to the end of the application.
19. In Postman, you should get a 200 response with your custom message.

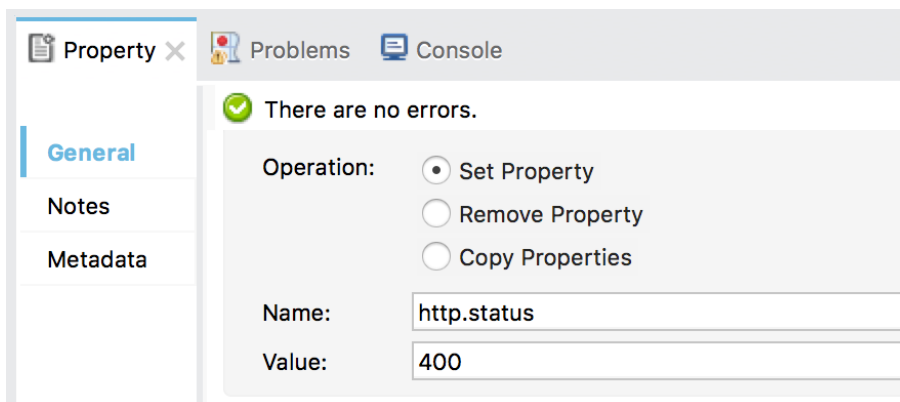


## Set the http status code

20. Return to getDeltaFlightsFlow in the Mule Design perspective.
21. In the catch exception strategy, add a Property transformer after the Set Payload transformer.

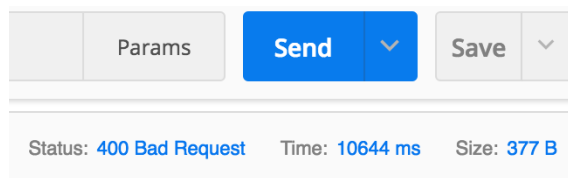


22. In the Property properties view, select Set Property.
23. Set the name to http.status and the value to 400.



## Test the application

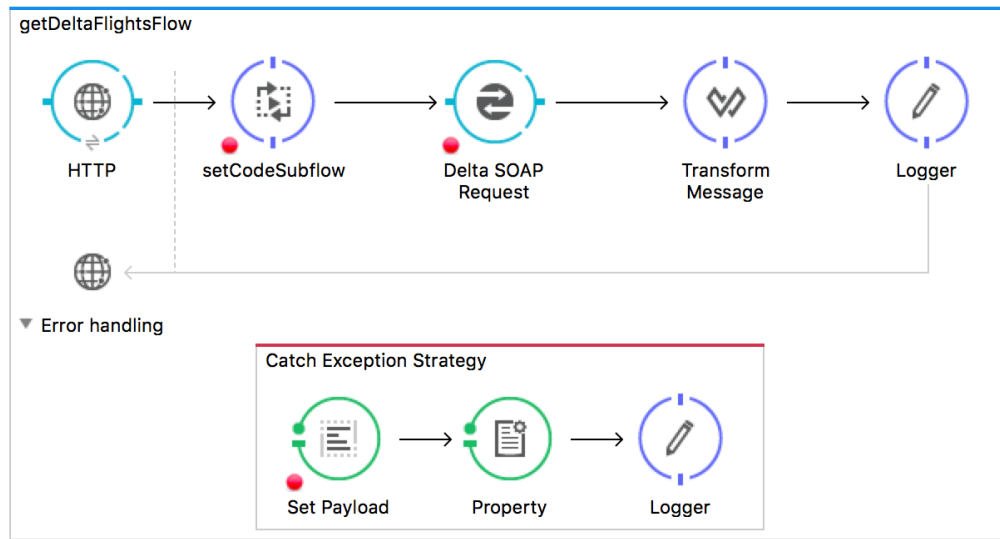
24. Redeploy the application.
25. In Postman, make another request to <http://localhost:8081/delta?code=FOO>.
26. In the Mule Debugger, step through the application.
27. Return to Postman; you should now get a 400 status code.



## Create a different type of error

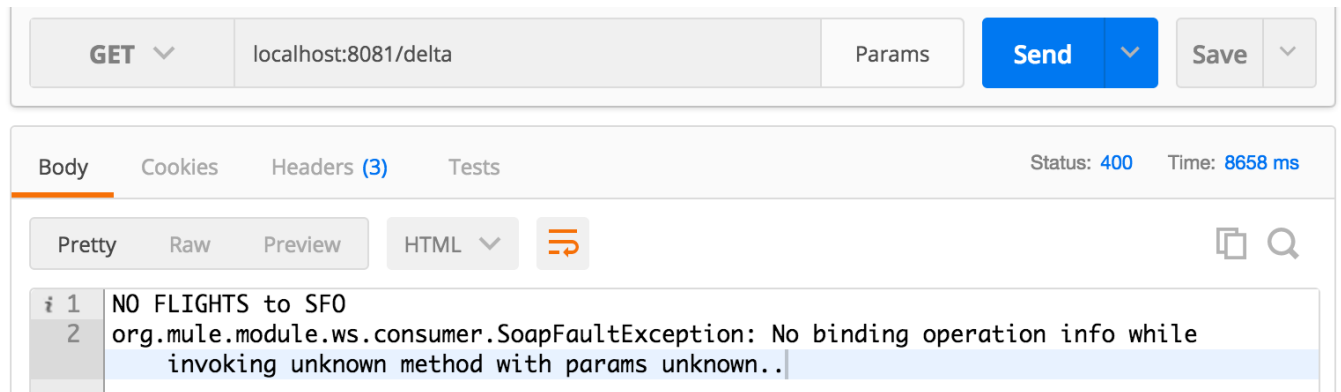
28. Return to getDeltaFlightsFlow.

29. Delete the Pass code processor.



## Test the application

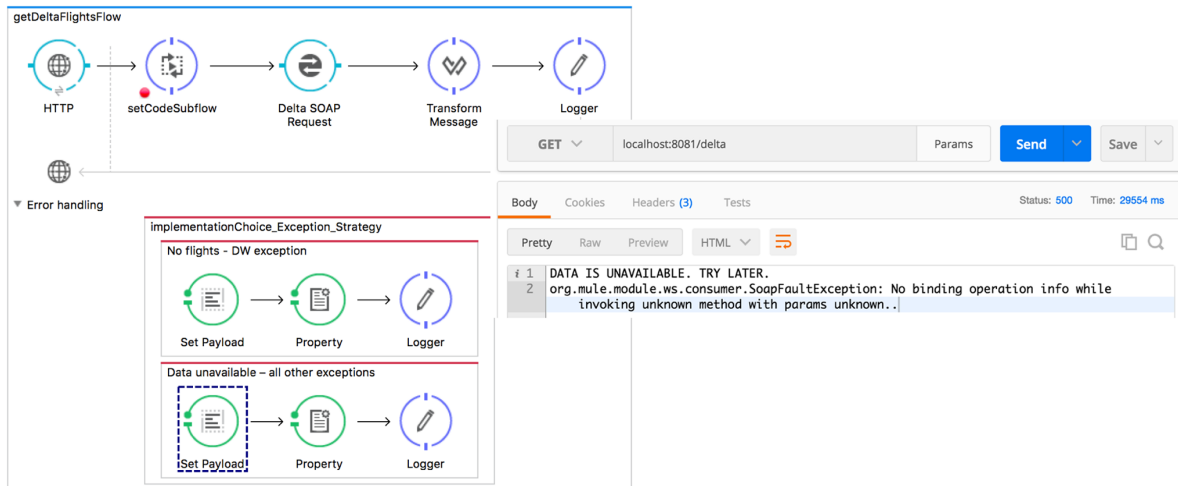
30. Redeploy the application.
31. In Postman, delete the query parameter and send the request.
32. In the Mule Debugger, step through the application until you get the exception and move into the exception handler; you should see the exception is handled by the same exception handler.
33. Drill-down into the Exception object in the debugger.
34. Click Resume.
35. In Postman, you should get the same message, but this time saying there are no flights to SFO even though that was not the problem.



## Walkthrough 9-2: Handle different types of messaging exceptions

In this walkthrough, you handle multiple types of exceptions thrown by the Delta flow. You will:

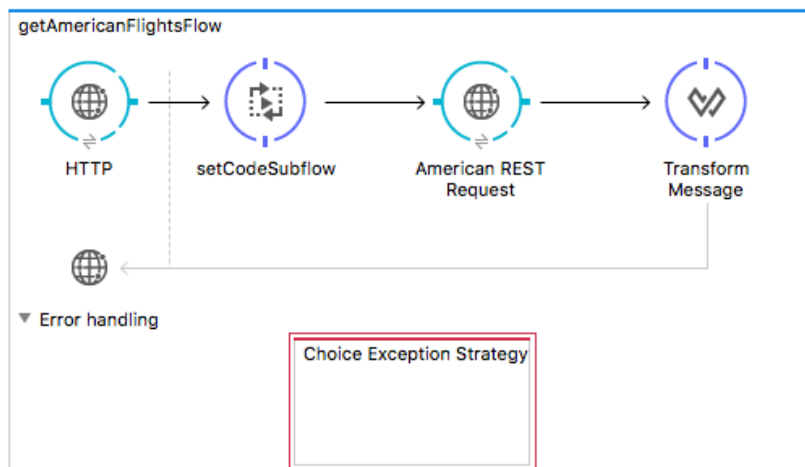
- Add and configure a choice exception strategy.
- Get exceptions handled by both of the catch exception strategies in the choice strategy.
- Create a new flow that calls a flow that has an exception so the exception can bubble up and be handled by the calling flow.



### Add a choice exception strategy

1. Return to `implementation.xml` in the Mule Design perspective.
2. Drag a `Choice Exception Strategy` from the Mule Palette and drop it into the error handling section of `getAmericanFlightsFlow`.

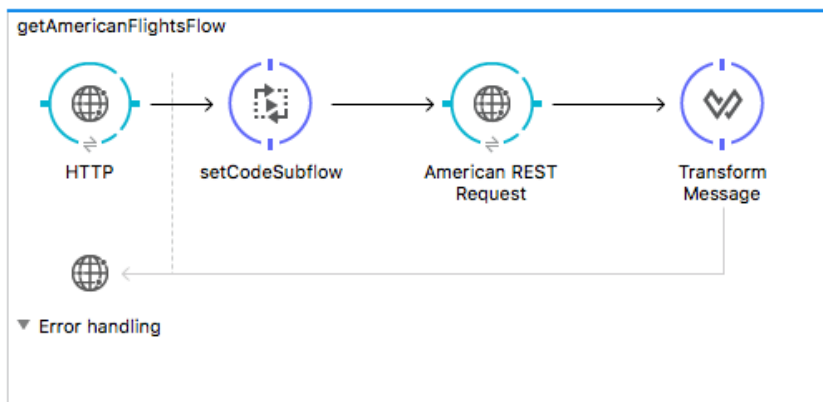
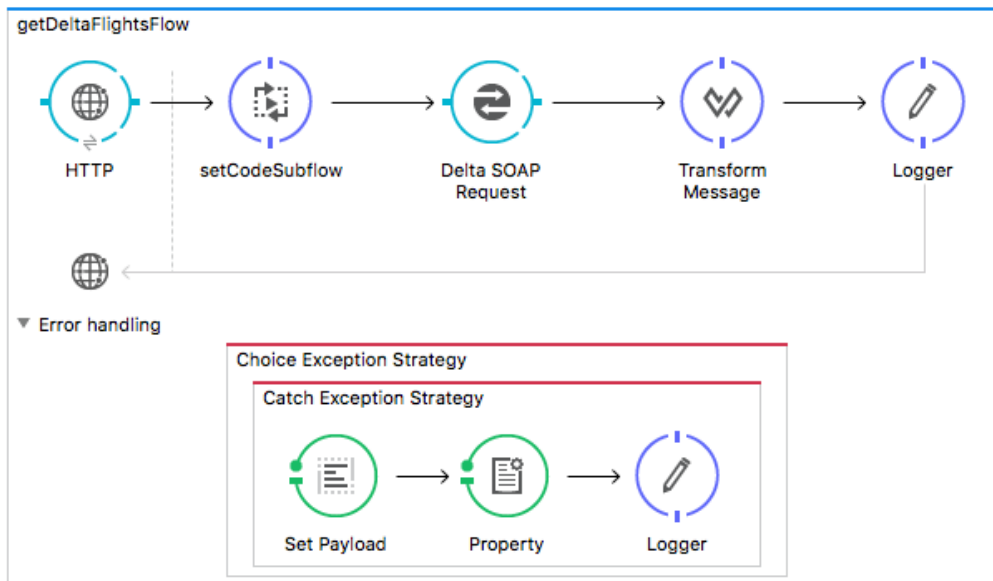
*Note: You cannot it directly to `getDeltaFlightsFlow` because that flow already contains an exception strategy, so you are adding it to `getAmericanFlightsFlow` temporarily.*



3. Drag the catch exception strategy from getDeltaFlightsFlow and drop it in the choice exception strategy in getAmericanFlightsFlow.
4. Drag getDeltaFlightsFlow and drop it above getAmericanFlightsFlow.

*Note: If you must scroll to see all the flows on the canvas, it is difficult to drag-and-drop elements near the bottom of the canvas.*

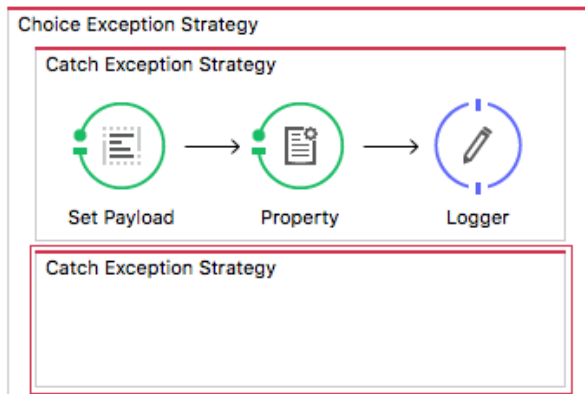
5. Drag the choice exception strategy from getAmericanFlightsFlow to getDeltaFlightsFlow.



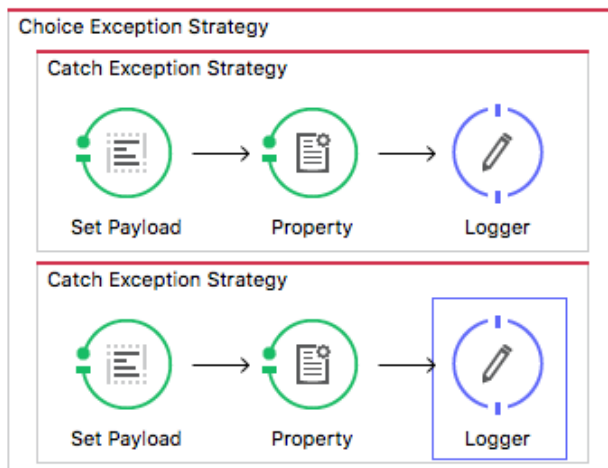


6. Drag a second Catch Exception Strategy from the Mule Palette and drop it in the choice exception strategy.

*Note: If you are having difficulty adding it, try dropping it on the catch exception strategy already there.*



7. Add a Set Payload transformer, a Property transformer, and a Logger to the new catch exception strategy.



8. In the Set Payload properties view, set the value the following MEL expression:  
`DATA IS UNAVAILABLE. TRY LATER. #['\n' + exception]`
9. In the Property properties view, select Set Property.
10. Set the name to `http.status` and the value to `500`.

## Configure the choice exception strategy

11. In the Properties view for the first catch exception strategy, set the display name to No flights – DW exception.
12. Set the execute when value to an expression for when a `com.mulesoft.weave.*` exception is thrown; use the `causeMatches()` method.

```
#[exception.causeMatches('com.mulesoft.weave.*')]
```

Catch Exception Strategy X Problems Console

There are no errors.

**General**

Notes

Display Name: No flights - DW exception

Settings

Configure conditional execution using an expression

Execute When: `#[exception.causeMatches('com.mulesoft.weave.*')]`

☒ Enable Notifications

13. In the Properties view for the second catch exception strategy, set the display name to Data unavailable – all other exceptions.
14. Leave the execute when value blank.

Data is unavailable - all other exceptions X Problems Console Mule Debugger

There are no errors.

**General**

Notes

Display Name: Data unavailable - all other exceptions

Settings

Configure conditional execution using an expression

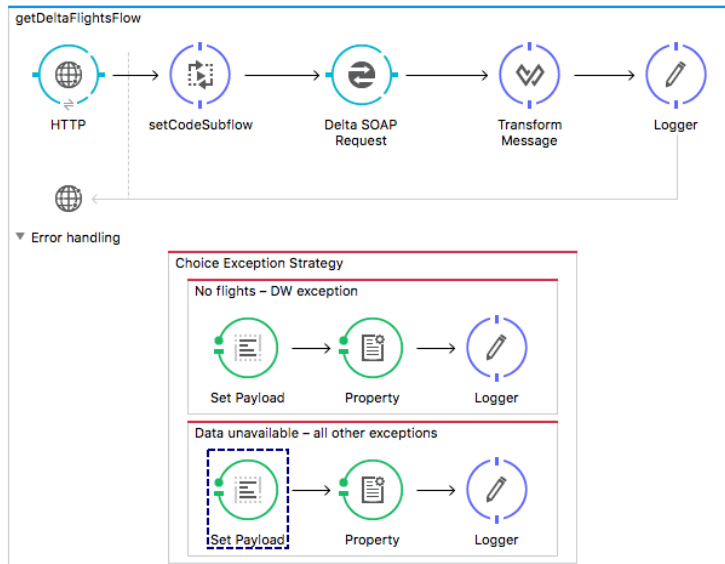
Execute When:

☒ Enable Notifications

## Test the application

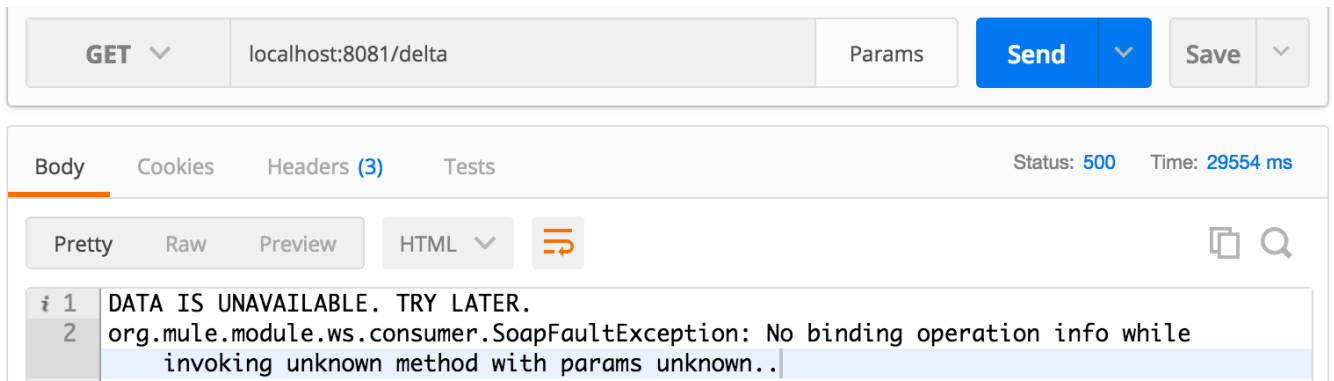
15. Debug the application.
16. In Postman, make another request to <http://localhost:8081/delta>.

17. In the Mule Debugger, step through the application; the exception should be handled by the second catch block.



18. Step to the end of the application.

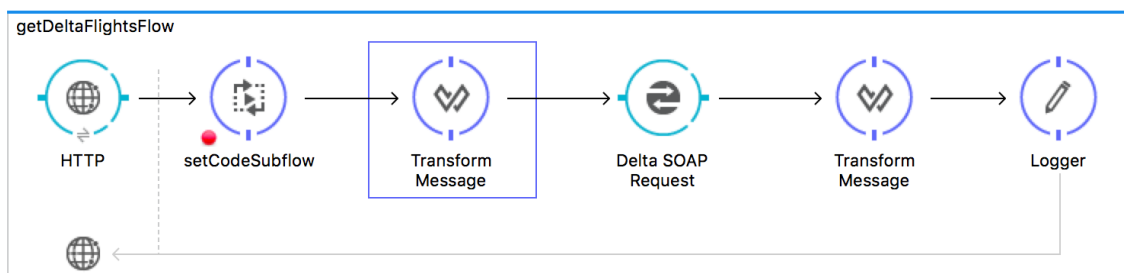
19. Return to Postman; you should see the second error message.



## Fix the request error

20. Return to `getDeltaFlightsFlow` and switch perspectives.

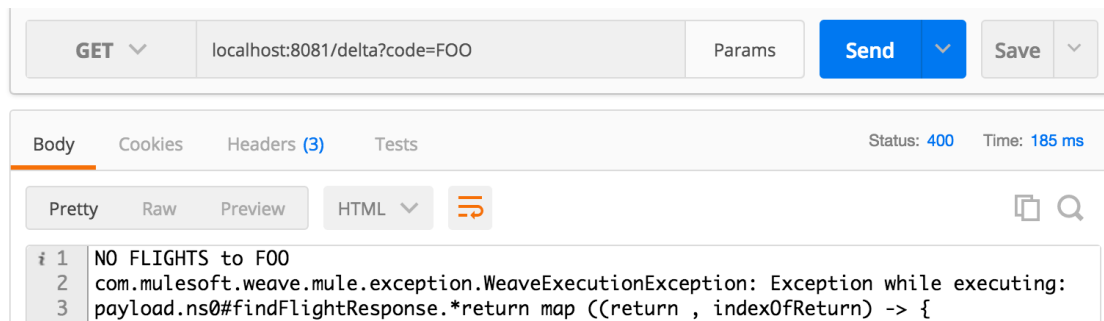
21. Add a `Transform Message` component before the `Delta SOAP Request` processor.



22. In the Transform Message properties view, drag the code flow variable in the input section to the destination element in the output section.

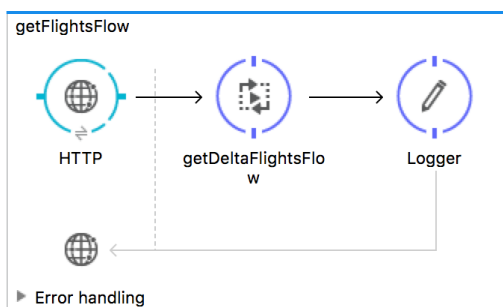
## Test the application

23. Run the project.
24. In Postman, make the same request to <http://localhost:8081/delta>; you should get the SFO results again.
25. Make a request to <http://localhost:8081/delta?code=FOO>; the exception is handled by the first exception strategy and you should get the NO FLIGHTS TO FOO message.



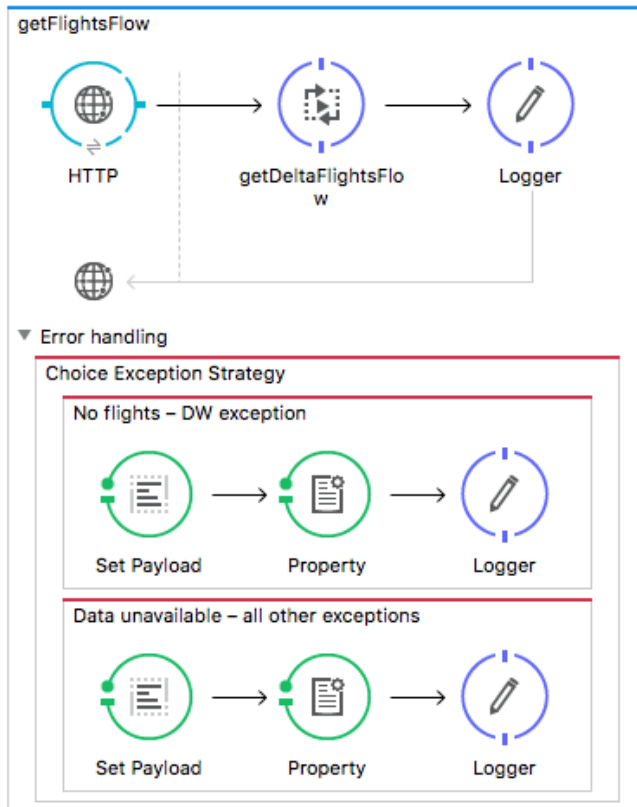
## Call the Delta flow from another flow

26. Return to Anypoint Studio.
27. From the Mule Palette, drag an HTTP connector and drop it at the top of the canvas above all the other flows.
28. In the HTTP properties view, set the connector configuration to the existing HTTP\_Listener\_Configuration.
29. Set the path to /flights and the allowed methods to GET.
30. Change the flow name to getFlightsFlow.
31. Add a Flow Reference component to the flow.
32. In the Flow Reference properties view, set the flow to getDeltaFlightsFlow.
33. Add a Logger to the end of the flow.



## Move a catch exception strategy to the calling flow

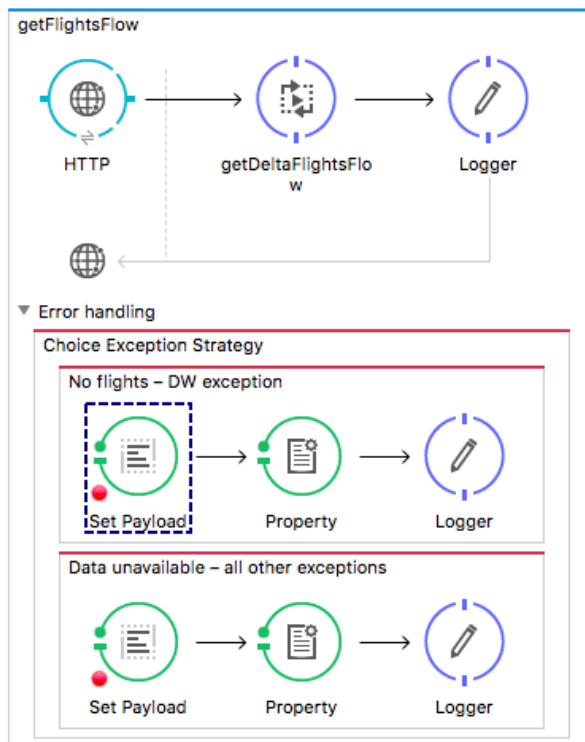
34. Expand the error handling section of getFlightsFlow.
35. Move the choice exception strategy from getDeltaFlightsFlow to getFlightsFlow.



36. Test the application. Make sure there is a breakpoint on each of the Set Payload transformers in the catch exception strategies.
37. Debug the project.
38. In Postman, make a request to <http://localhost:8081/flights?code=FOO>.

*Note: Call the new /flights endpoint; not the /delta one.*

39. In the Mule Debugger, step through the application; you should see the exception thrown by the Delta flow is caught by the calling flow.

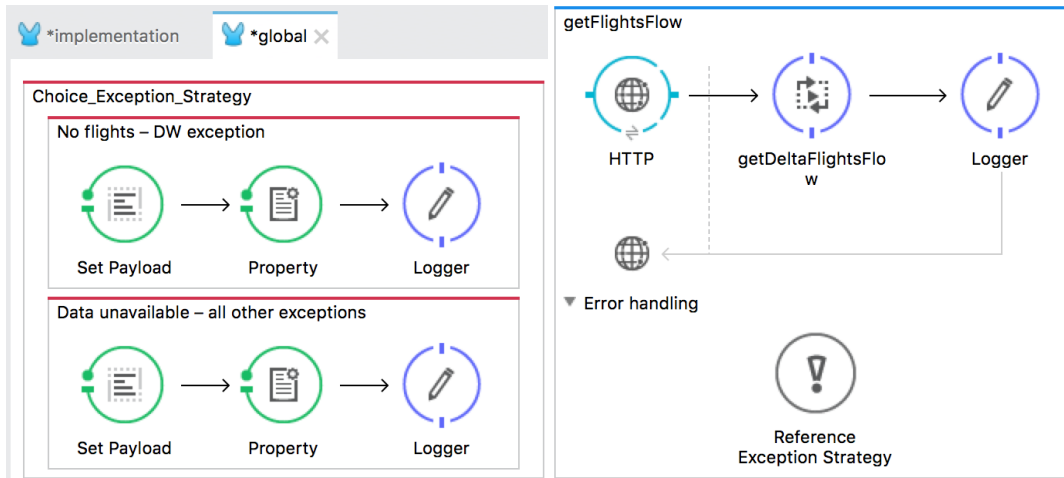


40. Step through the rest of the application and then stop the project and switch perspectives.

## Walkthrough 9-3: Create and use global exception strategies

In this walkthrough, you create a global exception handler for the project. You will:

- Create a global exception handler.
- Reference and use the global exception handler in flows.



### Create a global exception handler

1. Return to implementation.xml.
2. Switch to the Configuration XML view.
3. Locate the choice exception strategy and select it and cut it.

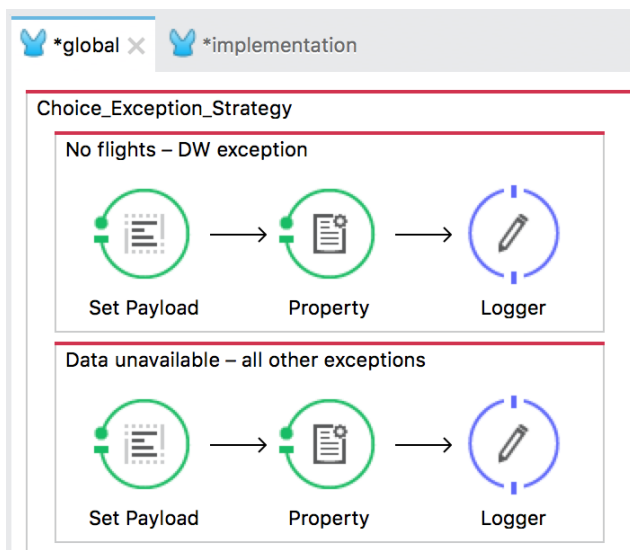
```
global implementation
12 <flow name="getFlightsFlow">
13   <http:listener config-ref="HTTP_Listener_Configuration" path="/flights" allowedMethod="GET"/>
14   <flow-ref name="getDeltaFlightsFlow" doc:name="getDeltaFlightsFlow"/>
15   <logger level="INFO" doc:name="Logger"/>
16   <choice-exception-strategy doc:name="Choice Exception Strategy">
17     <catch-exception-strategy when="#[exception.causeMatches('com.mulesoft.weave.*')]">
18       <set-payload value="NO FLIGHTS to #[flowVars.code + '\n' + exception]" doc:name="Set Payload"/>
19       <set-property propertyName="http.status" value="400" doc:name="Property"/>
20       <logger level="INFO" doc:name="Logger"/>
21     </catch-exception-strategy>
22     <catch-exception-strategy doc:name="Data unavailable &#211; all other exceptions">
23       <set-payload value="DATA IS UNAVAILABLE. TRY LATER. #[\n' + exception]" doc:name="Set Payload"/>
24       <set-property propertyName="http.status" value="500" doc:name="Property"/>
25       <logger level="INFO" doc:name="Logger"/>
26     </catch-exception-strategy>
27   </choice-exception-strategy>
28 </flow>
29 <sub-flow name="setCodeSubflow">
```

4. Go to the Configuration XML view in global.xml.
5. Place the cursor on a new line inside the start and end mule tags after the other tags.
6. Paste the choice exception strategy.

7. In the choice-exception-strategy tag, change doc:name to name and replace the spaces in the name with underscores; the problem should go away

```
*global x *implementation
16 </http:request-config>
17 <ws:consumer-config name="Delta_Web_Service_Consumer" wsdlLocation="http://mu.mulesoft-train
18
19 <choice-exception-strategy name="Choice_Exception_Strategy">
20 <catch-exception-strategy when="#[exception.causeMatches('com.mulesoft.weave.*')]" c
21 <set-payload value="NO FLIGHTS to #[flowVars.code + '\n' + exception]" doc:name
22 <set-property propertyName="http.status" value="400" doc:name="Property"/>
23 <logger level="INFO" doc:name="Logger"/>
24 </catch-exception-strategy>
25 <catch-exception-strategy doc:name="Data unavailable &#8211; all other exceptions">
26 <set-payload value="DATA IS UNAVAILABLE. TRY LATER. #['\n' + exception]"
27 " doc:name="Set Payload"/>
28 <set-property propertyName="http.status" value="500" doc:name="Property"/>
29 <logger level="INFO" doc:name="Logger"/>
30 </catch-exception-strategy>
31 </choice-exception-strategy>
32
33 </mule>
```

8. Switch to the Message Flow view; you should see the choice exception strategy.

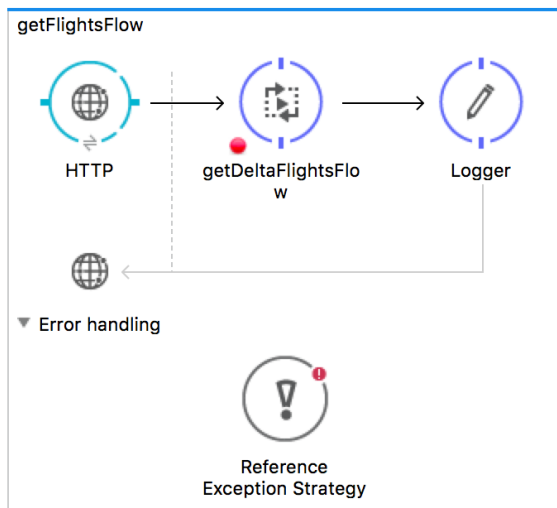


## Use the global exception handler

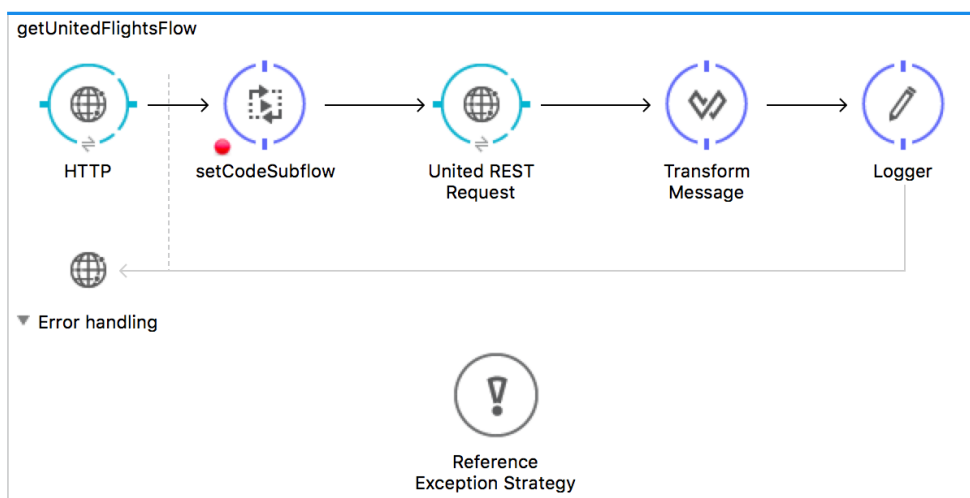
9. Return to implementation.xml and switch to the Message Flow view.
10. Locate getFlightsFlow and expand its error handling section.



11. Drag a Reference Exception Strategy from the Mule Palette and drop it in the error handling section.



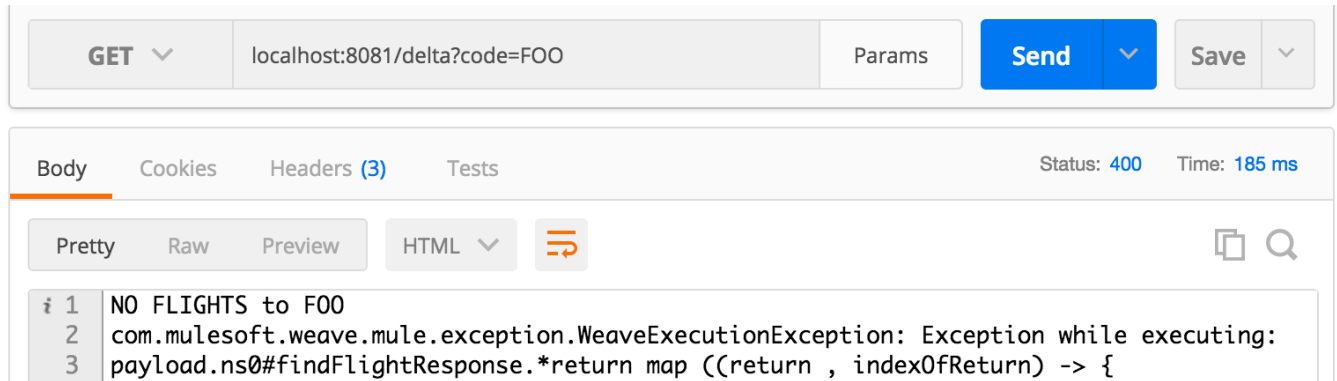
12. In the Reference Exception Strategy properties view, set the global exception strategy to the existing Choice\_Exception\_Strategy.
13. Expand the error handling section of getUnitedFlightsFlow.
14. Drag a Reference Exception Strategy from the Mule Palette and drop it in the error handling section.
15. In the Properties view, set the global exception strategy to Choice\_Exception\_Strategy.



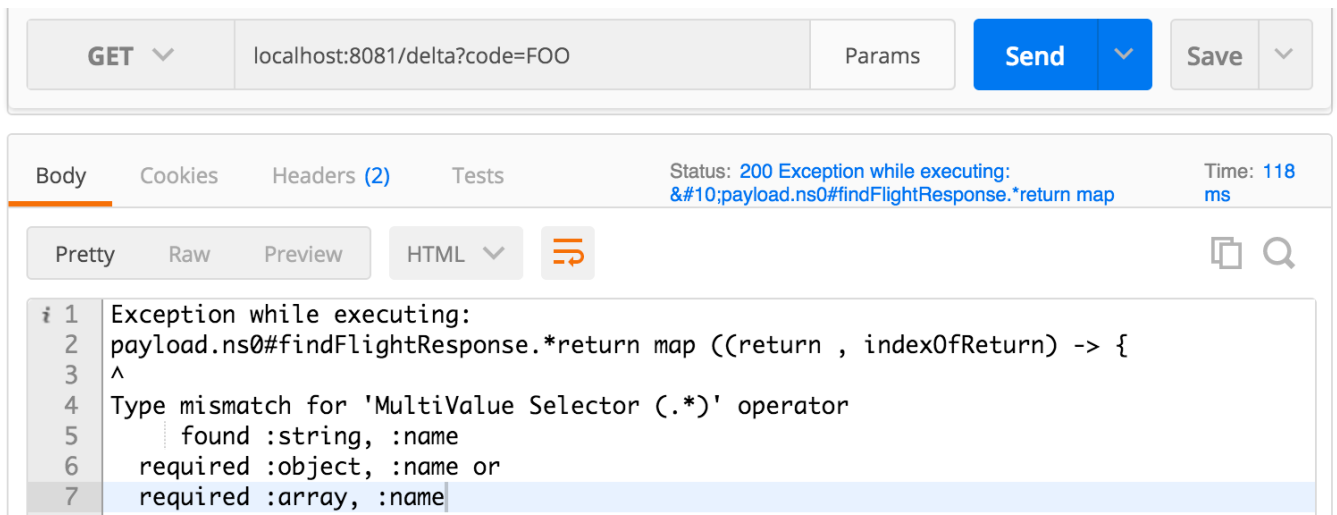
*Note: You could add a reference exception strategy to the rest of the flows, but instead, you will create a global default exception strategy in the next walkthrough.*

## Test the application

16. Run the project.
17. In Postman, make the same request to <http://localhost:8081/flights?code=FOO>; you should get the same NO FLIGHTS to FOO message.



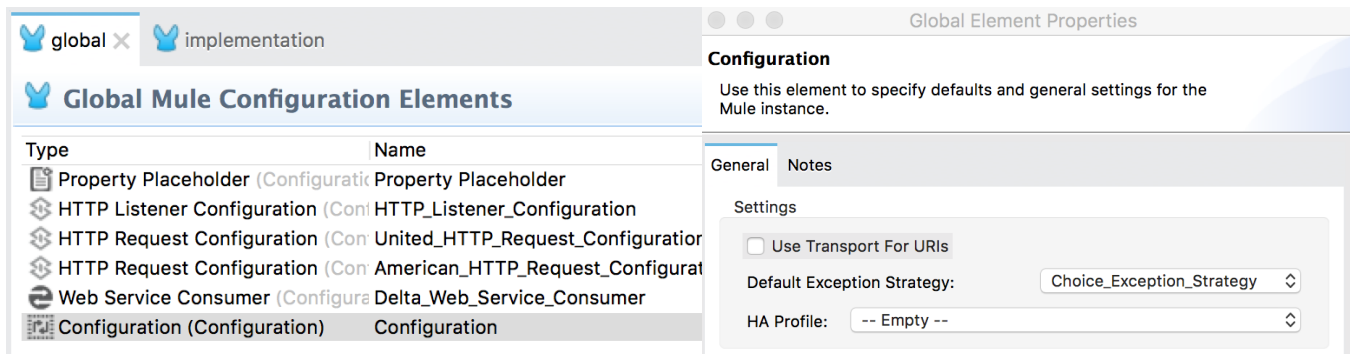
18. Make a request to <http://localhost:8081/united?code=FOO>; you should see the same message.
19. Make a request to <http://localhost:8081/delta?code=FOO>; you should NOT see the same message because the exception is not being handled.



## Walkthrough 9-4: Specify a global default exception strategy

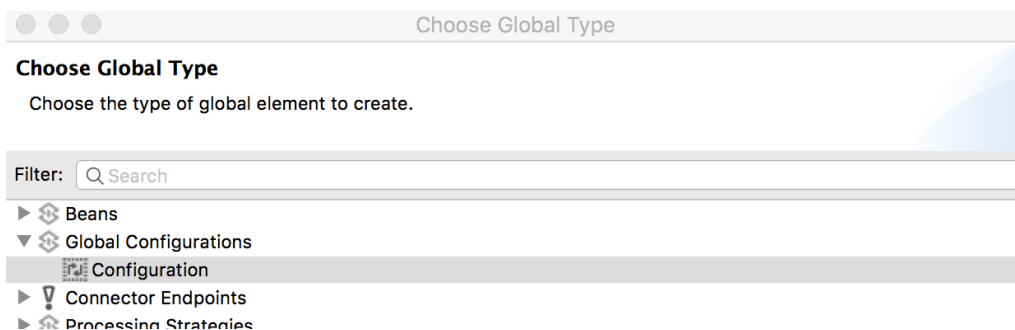
In this walkthrough, you change the default exception handling for the application. You will:

- Create a global configuration element in the global.xml file.
- Specify a default exception strategy in the global configuration element.
- Remove the existing exception handling strategies.
- Use the default exception handling strategy.

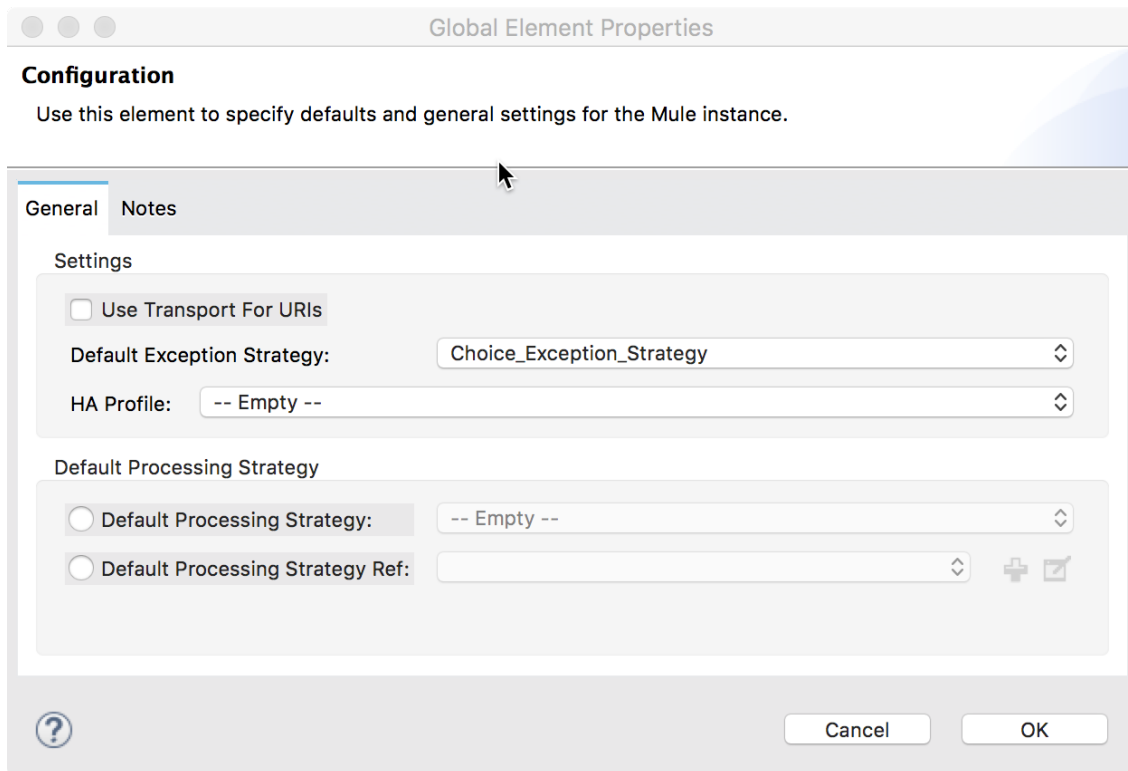


### Specify a global default exception strategy

1. Return to global.xml.
2. Switch to the Global Elements view.
3. Click the Create button.
4. In the Choose Global Type dialog box, select Global Configurations > Configuration and click OK.

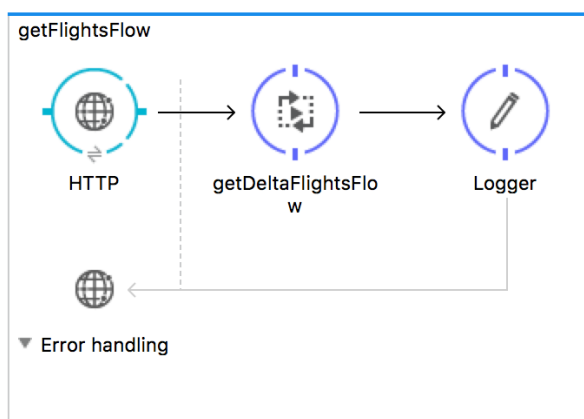


5. In the Global Element Properties dialog box, set the default exception strategy to `Choice_Exception_Strategy` and click OK.

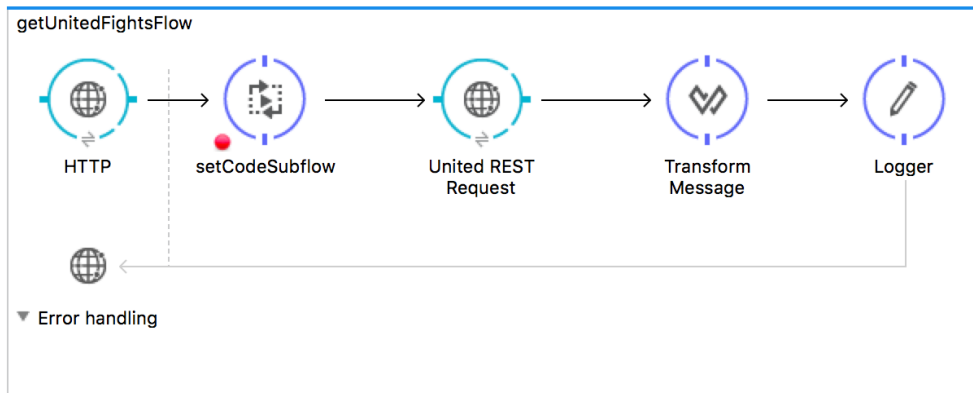


## Remove the existing exception strategy references

6. Return to `implementation.xml`.
7. Delete the reference exception strategy in `getFlightsFlow`.

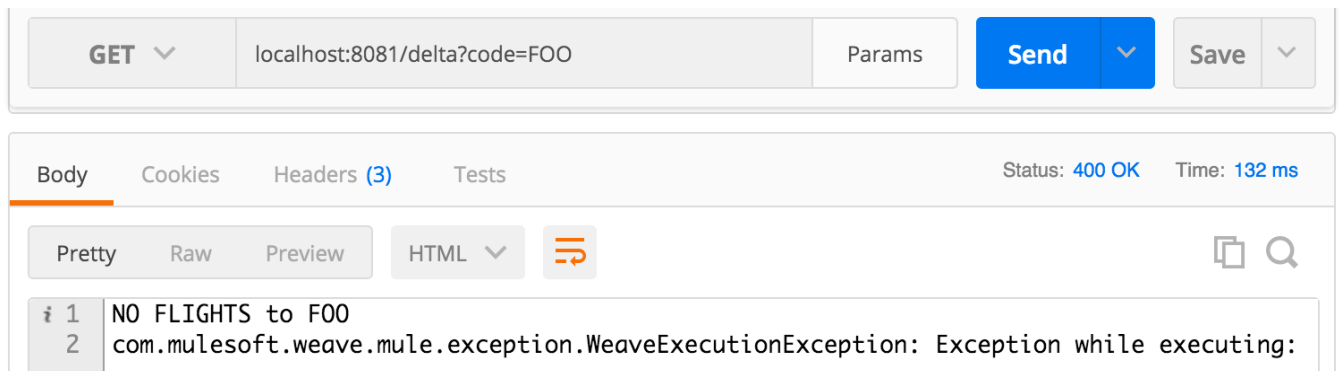


8. Delete the reference exception strategy in getUnitedFlightsFlow.



## Test the application

9. Save all the files and run the project.
10. In Postman, make a request to `http://localhost:8081/flights?code=FOO`; you should still see the no flights error message displayed.
11. Make a request to `http://localhost:8081/delta?code=FOO`; you should now see the no flights error message displayed.

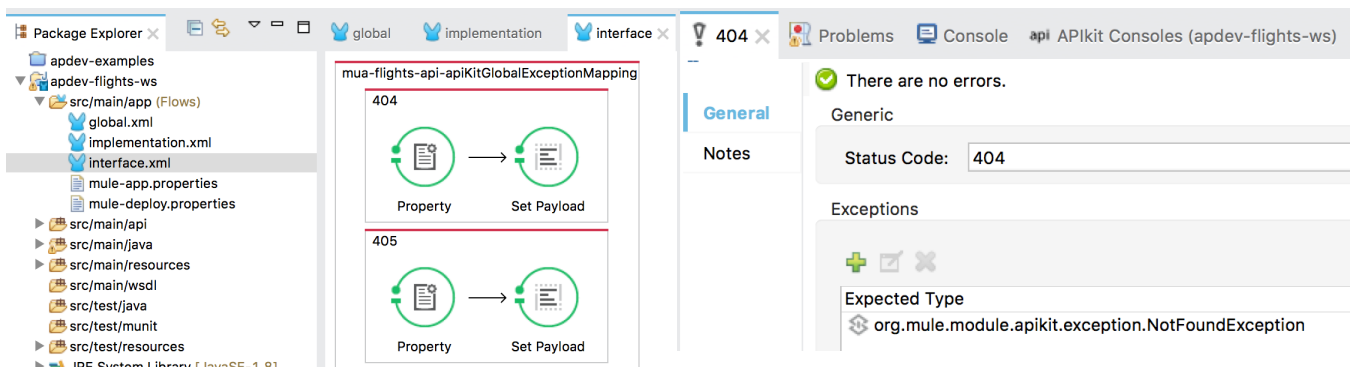


12. Return to Anypoint Studio and stop the project.

## Walkthrough 9-5: Review a mapping exception strategy

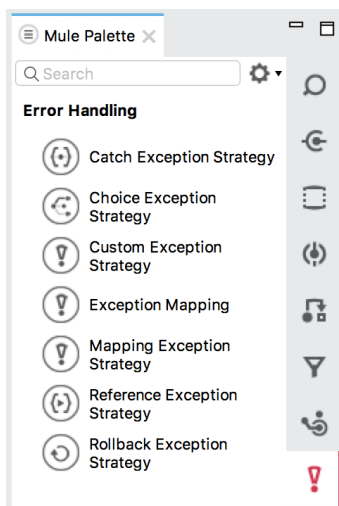
In this walkthrough, you review the mapping exception strategy that was created automatically by APIkit for the interface. You will:

- Locate the mapping exception strategy.
- Review the exception mappings.



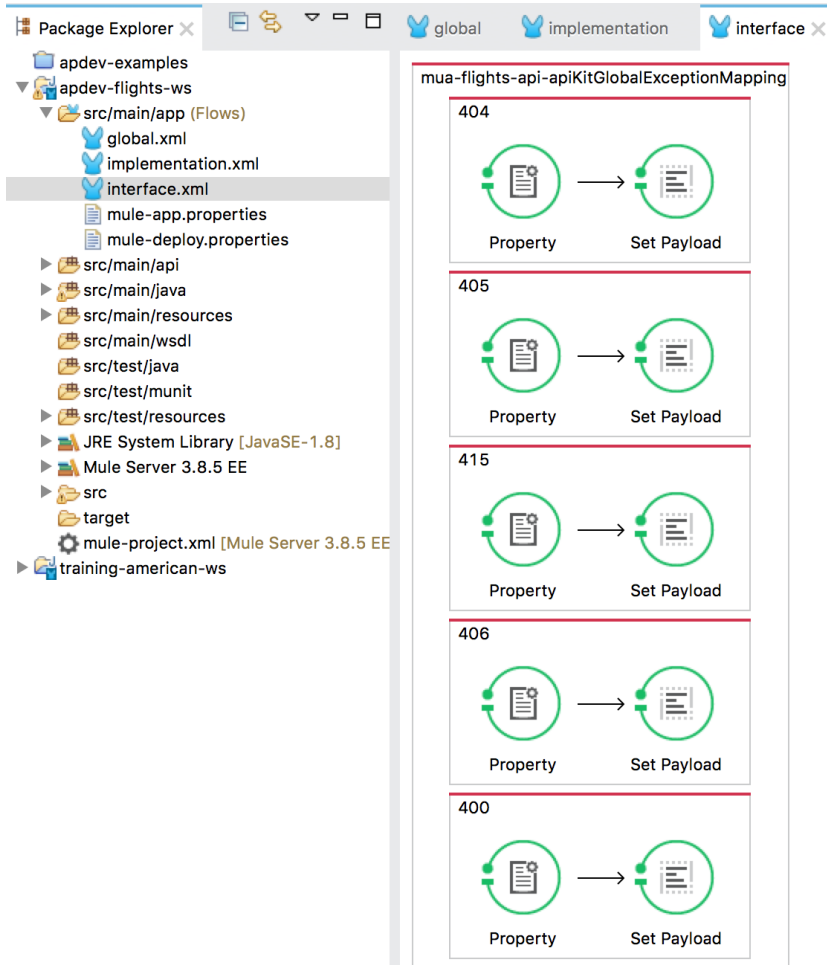
### Browse the error handling elements in the Mule Palette

1. In the apdev-flights-ws project, open interface.xml in src/main/app.
2. In the Mule Palette, select the Error Handling tab.
3. Locate the Mapping Exception Strategy.

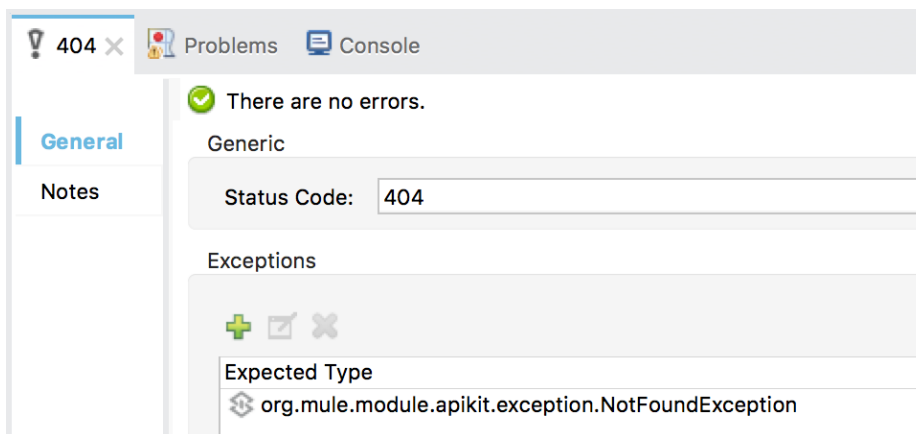


## Review a mapping exception strategy

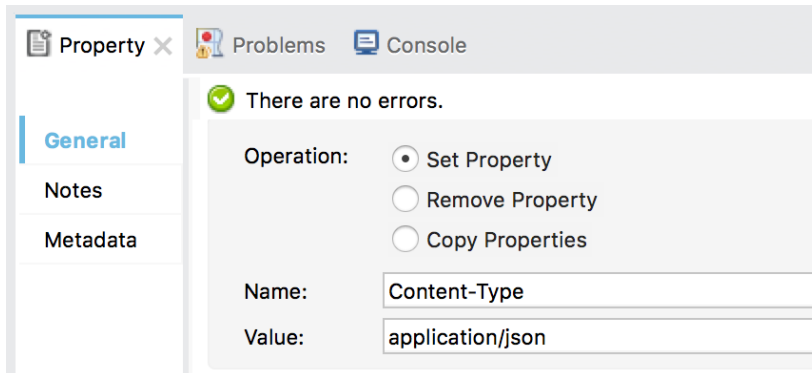
4. In interface.xml, locate the mua-flights-apiKitGlobalExceptionMapping Mapping exception strategy.



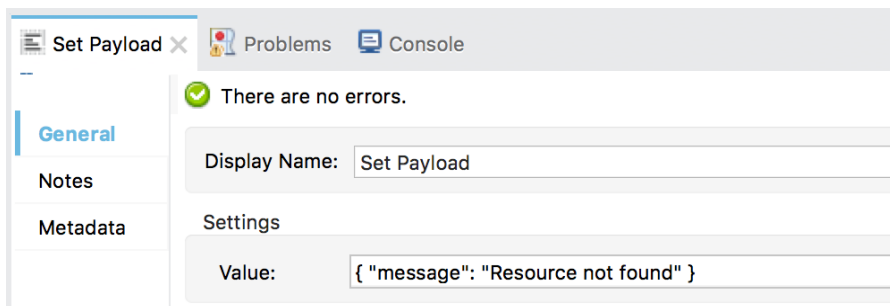
5. Double-click the 404 exception mapping and look at the settings in the 404 properties view.



- Double-click the Property transformer in the 404 exception mapping and look at the settings in the Set Payload properties view.



- Double-click the Set Payload transformer in the 404 exception mapping and look at the settings in the Set Payload properties view.



- Look at the values for the Set Payload transformers in the other exception mappings.