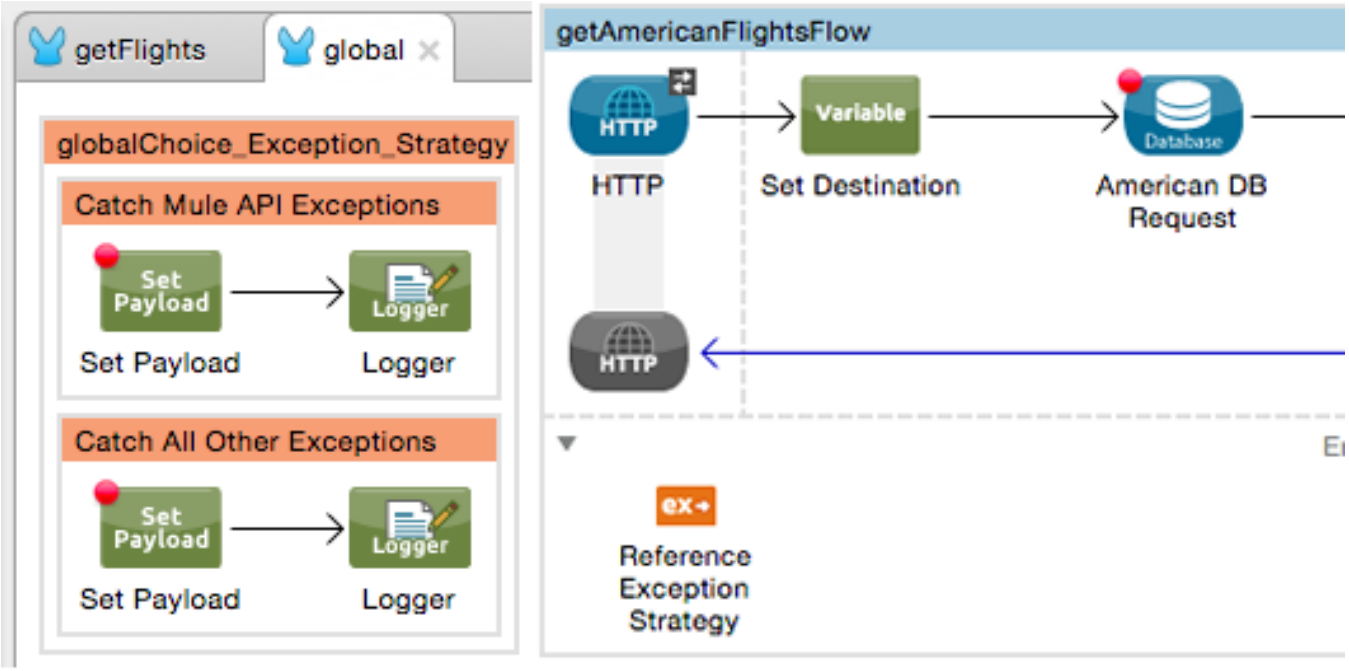
Module 8: Handling Errors



In this module, you will learn:

- About the different types of exception strategies.
- To handle messaging exceptions in flows.
- To create and use global exception handlers.
- To specify a global default exception strategy.

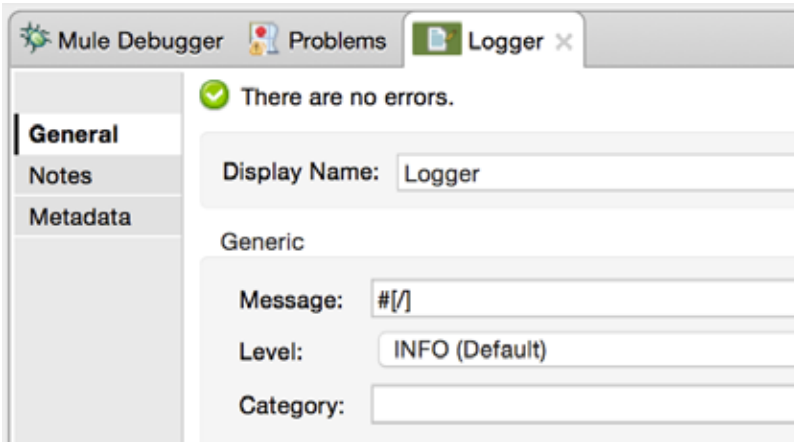
Walkthrough 8-1: Handle messaging exceptions

In this walkthrough, you will create errors in the MUA application and handle the exceptions that are thrown. You will:

- Create an error in the flight form flow.
- Catch the exception and send an error message back to the requester.
- Create database and web service request errors.
- Let the exceptions bubble up and be handled by the main flow.

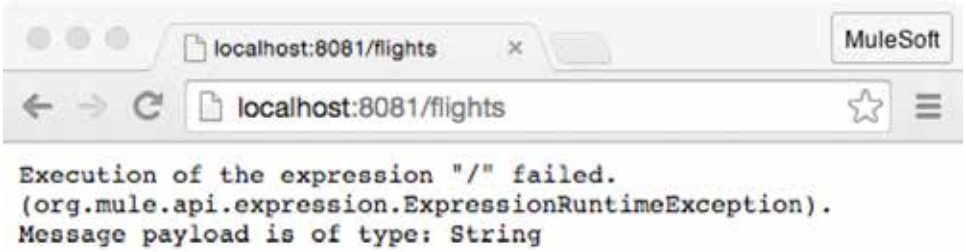
Create an error in the flight form flow

1. Return to getFlights.xml in apessentials.
2. Navigate to getFlightFormFlow.
3. Set the Logger message to an incorrect MEL expression, like #[/].



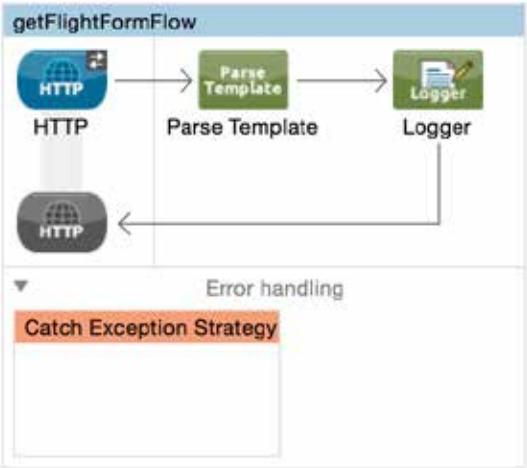
4. Run the application.

5. Make a request to http://localhost:8081/flights; you should get an error.

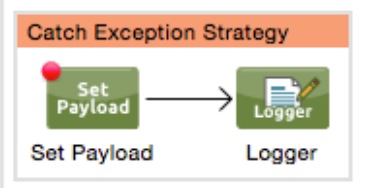


Add a catch exception strategy

6. Return to getFlightFormFlow and click the arrow to expand the error handling section.
7. Drag and drop a Catch Exception Strategy from the error handling section of the palette into the error handling section of the flow.



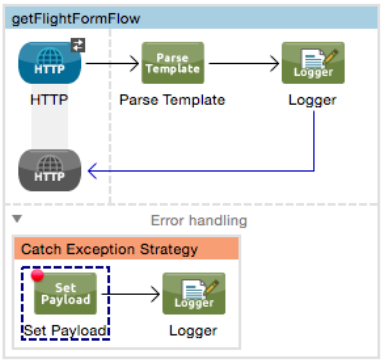
8. Add a Set Payload transformer to the Catch Exception Strategy.
9. In the Set Payload Properties view, set the value to The application is currently unavailable.
10. Add a breakpoint to the transformer.
11. Add a Logger after the transformer.



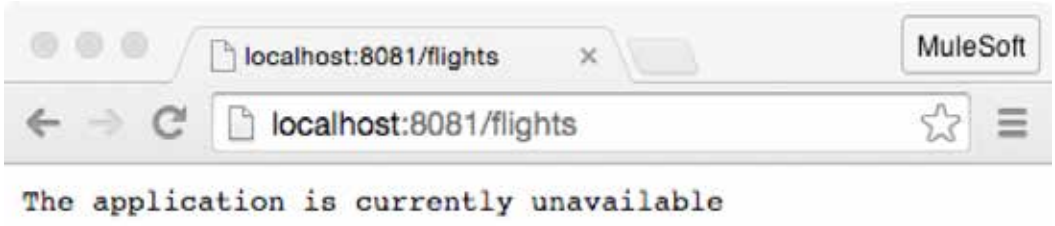
Test the application

12. Save the file and debug the application.
13. Refresh the browser window.

14. Step through the application; you should see the exception thrown in getFlightFormFlow and handled by the exception handler.



15. Step to the end of the application; you should see your message in the browser window.



Fix the flight flow

16. Remove the incorrect message expression in the Logger in getFlightFormFlow.

Create a database error

17. Navigate to the Database connector endpoint in getAmericanFlightsFlow.
18. Modify the name of the table in the database query; for example, change it to flight.

```
SELECT *
FROM flight
WHERE toAirport = #[flowVars.destination]
```

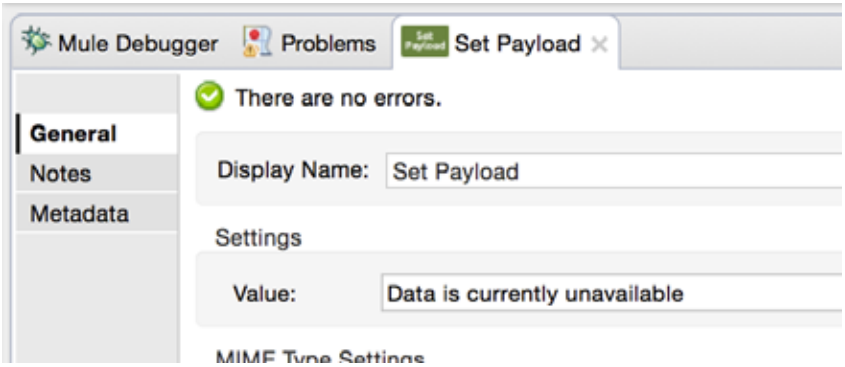
19. Click the Apply Changes button.
20. Ignore the Problem while fetching metadata popup window that appears; it will disappear on its own.
21. Run the application.
22. Refresh the browser window and submit the form for SFO and American; you should see no flights and no message.

23. Look at the console; you should see a MySQLSyntaxErrorException.

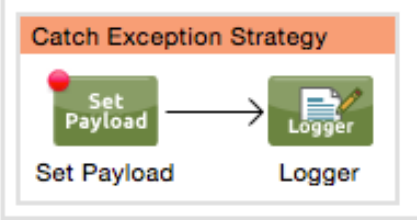
```
--------------------------------------------------------------------------------
Root Exception stack trace:
com.mysql.jdbc.exceptions.MySQLSyntaxErrorException: Table 'training.flight' doesn't exist
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:936)
        at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:2985)
        at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1631)
    + 3 more (set debug level logging or '-Dmule.verbose.exceptions=true' for everything)
********************************************************************************
```

Add a catch exception strategy to the main flow

24. Click the arrow to expand the error handling section of getFlightsFlow.
25. Add a Catch Exception Strategy into the error handling section of the flow.
26. Add a Set Payload transformer to the Catch Exception Strategy.
27. In the Set Payload Properties view, set the value to Data is currently unavailable.



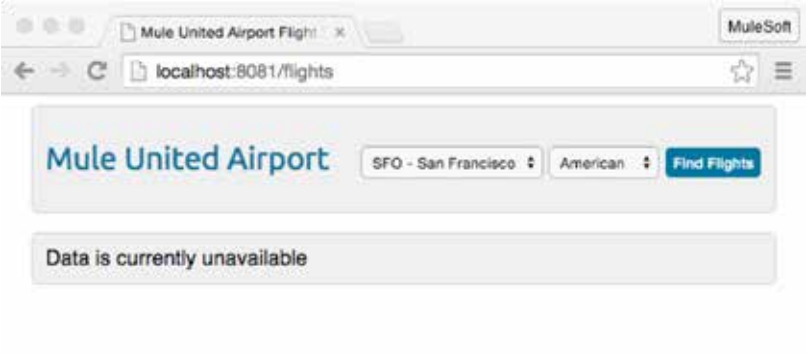
28. Add a breakpoint to the transformer.
29. Add a Logger after the Set Payload transformer.



Test the application

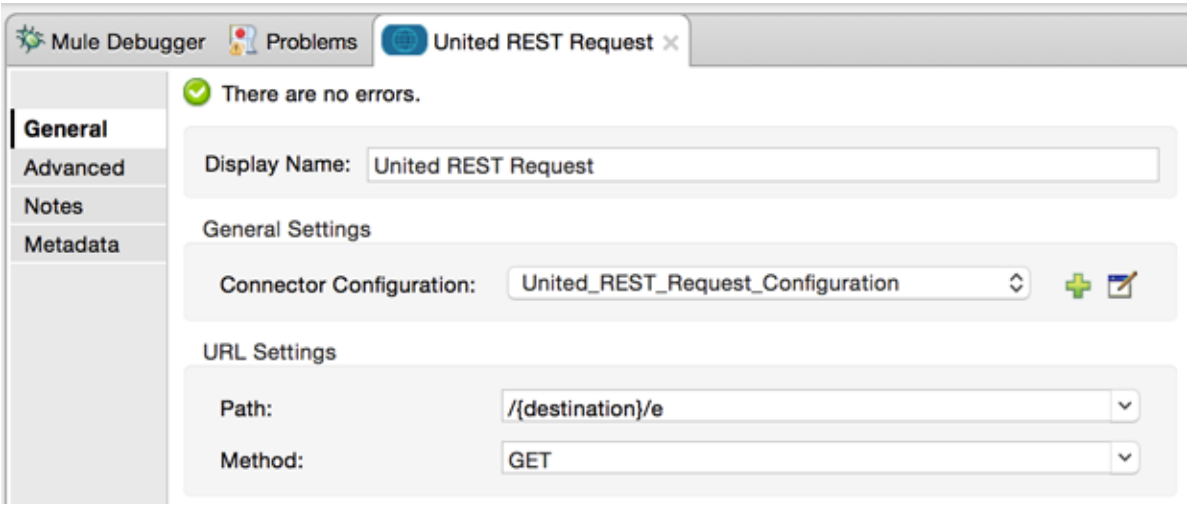
30. Save the file and debug the application.
31. Submit the form for SFO and American.
32. Step through the application; you should see the exception thrown in getAmericanFlightsFlow and handled by the exception handler in the main flow.

33. Return to the browser window; you should see your error message.



Make a RESTful web service request error

34. Navigate to the United REST Request endpoint in getUnitedFlightsFlow.
35. Change the path to an invalid path, like /{destination}/e.

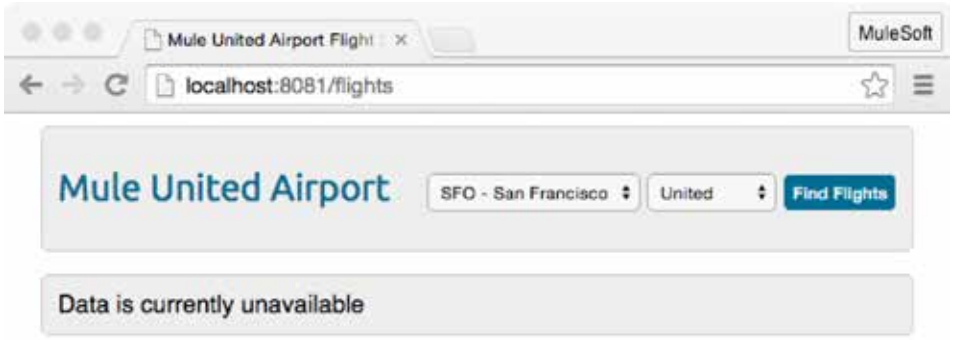


Note: You can also modify the connector configuration to use an invalid host, port, or path to make a request error.

Test the application

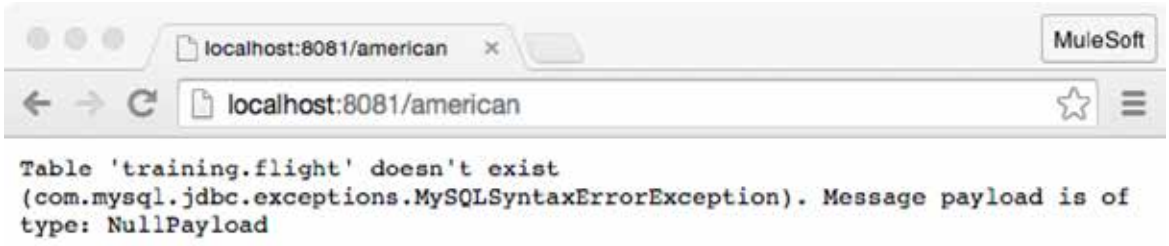
36. Save the file to redeploy the application.
37. Refresh the browser window and submit the form for SFO and United.

38. Step through the application; you should see the exception thrown in the getUnitedFlightsFlow is also handled by the exception handler in the main flow.

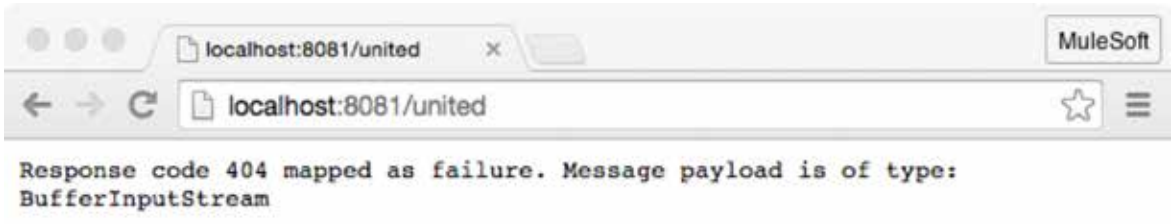


Test calls to the airline flows

39. Make a request to http://localhost:8081/american and step through the application; you should get a SQL exception – the exception is not handled.



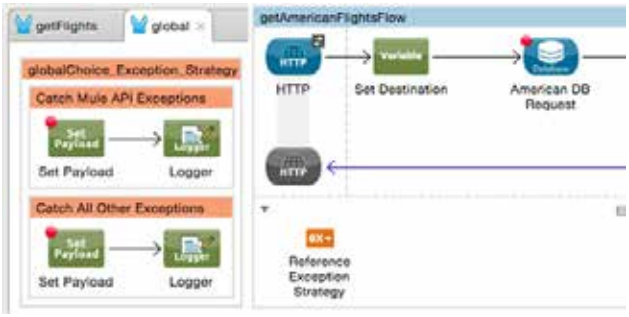
40. Make a request to http://localhost:8081/united and step through the application; you should get an exception in the console and an error in the browser window.



Walkthrough 8-2: Create and use global exception handlers

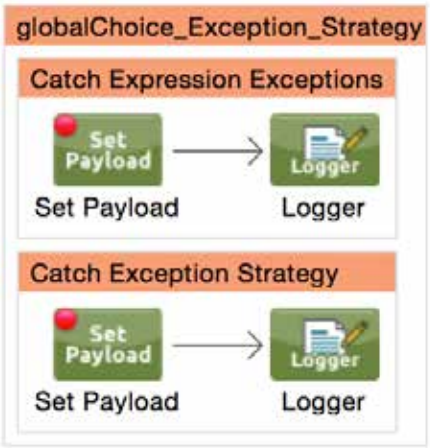
In this walkthrough, you will create global exception handlers in the MUA application. You will:

- Create a global exception handler.
- Use the Choice Exception Strategy.
- Use an expression to reference the type of exception thrown.
- Reference and use the global exception handler in the airline flows.



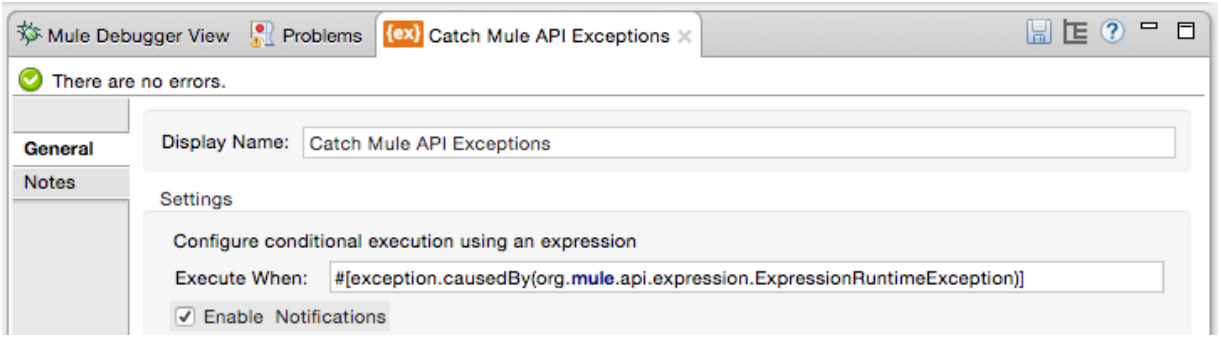
Create a global exception handler

1. Open global.xml in apessentials and go to the Message Flow view.
2. Drag a Choice Exception Strategy to the canvas.
3. Drag a Catch Exception Strategy and drop it in the Choice Exception Strategy.
4. Add a second Catch Exception Strategy.
5. Add a Set Payload transformer to each Catch Exception Strategy.
6. Add a Logger to each Catch Exception Strategy.
7. Add a breakpoint to each Set Payload transformer.
8. In the Properties view for the first Catch Exception Strategy, set the name to Catch Expression Exceptions.

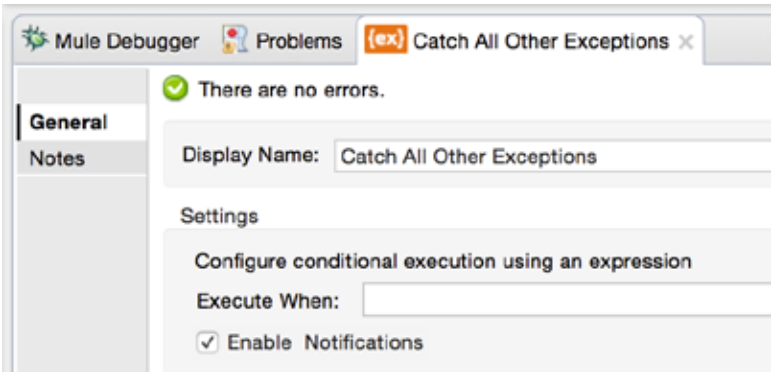


9. Write an execute when expression to use the exception.causeMatches() method to catch any org.mule.api exceptions.

```
#[exception.causedBy(org.mule.api.expression.ExpressionRuntimeException)]
```



10. In the Properties view for the second Catch Exception Strategy, set the name to Catch All Other Exceptions; do not set an execute when expression.

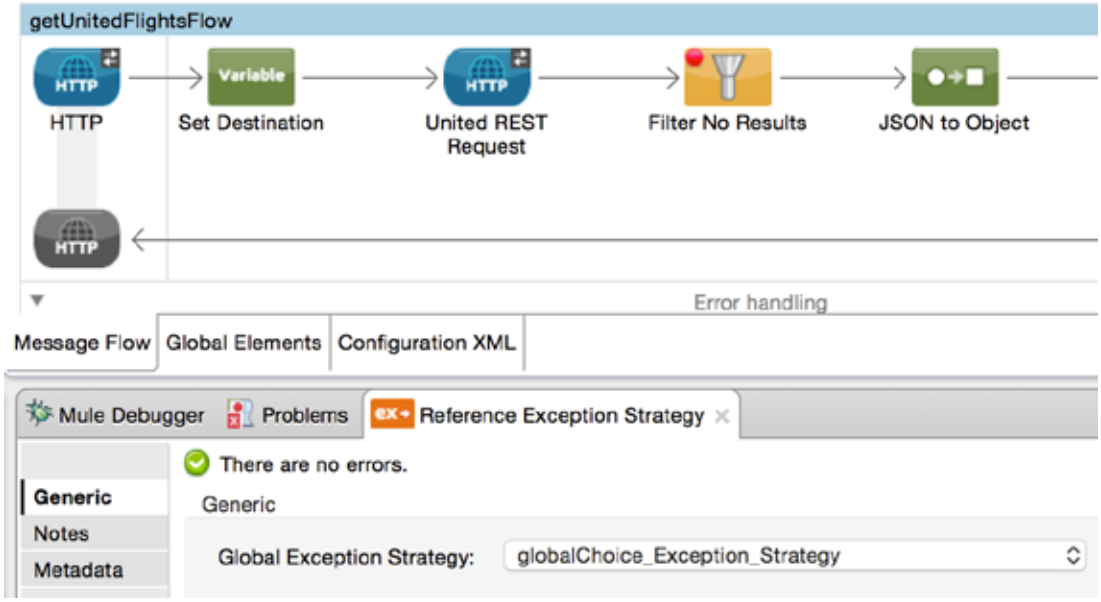


11. Double-click the Set Payload transformer in the Catch Expression Exceptions strategy.
12. In the Properties view, set the value to The application is currently unavailable.
13. Double-click the Set Payload transformer in the Catch All Other Exceptions strategy.
14. In the Properties view, set the value to Data is currently unavailable.
15. Save the file.

Use the global exception handler

16. Return to getFlights.xml.
17. Locate getUnitedFlightsFlow and expand its error handling section.
18. Drag a Reference Exception Strategy from the palette and drop it in the error handling section.

19. In the Properties view, set the global exception strategy to globalChoice_Exception_Strategy.



20. Drag a Reference Exception Strategy to the error handling section of getAmericanFlightsFlow.
21. In the Properties view, set the global exception strategy to globalChoice_Exception_Strategy.
22. Delete the existing Catch Strategy Exception in getFlightsFlow.
23. Replace it with a Reference Exception Strategy with the global exception strategy set to globalChoice_Exception_Strategy.
24. Delete the existing Catch Strategy Exception in getFlightFormFlow.
25. Replace it with a Reference Exception Strategy with the global exception strategy set to globalChoice_Exception_Strategy.

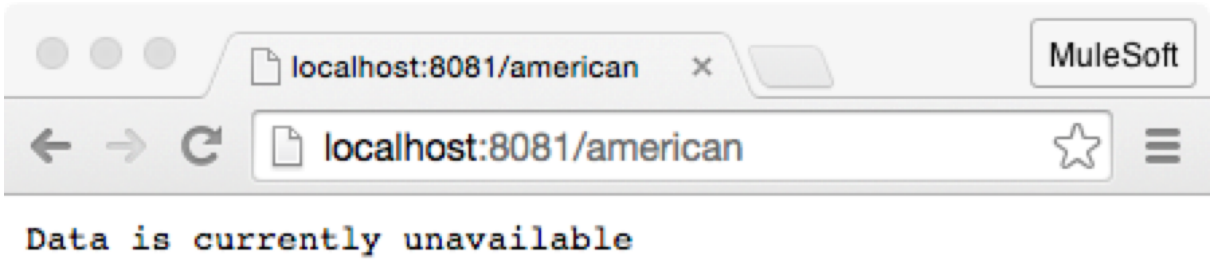
    Note: You could add a Reference Exception Strategy to the rest of the flows, but instead, you will create a global default exception strategy in the next walkthrough.

Create an error in the flight form flow again

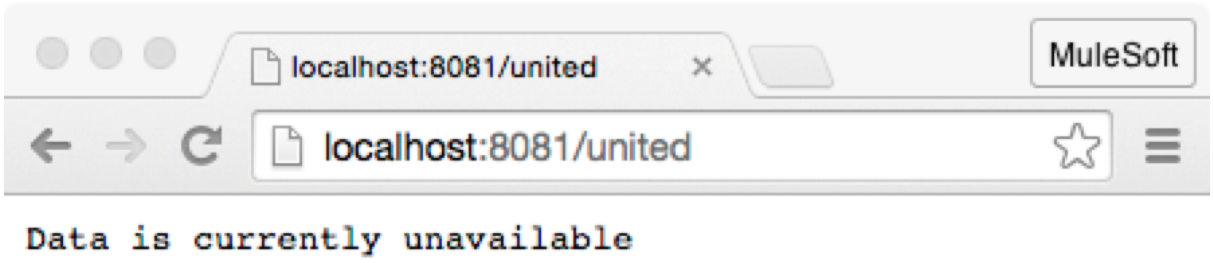
26. In getFlightFormFlow, set the Logger to an incorrect MEL expression again, like #[/].

Test the application

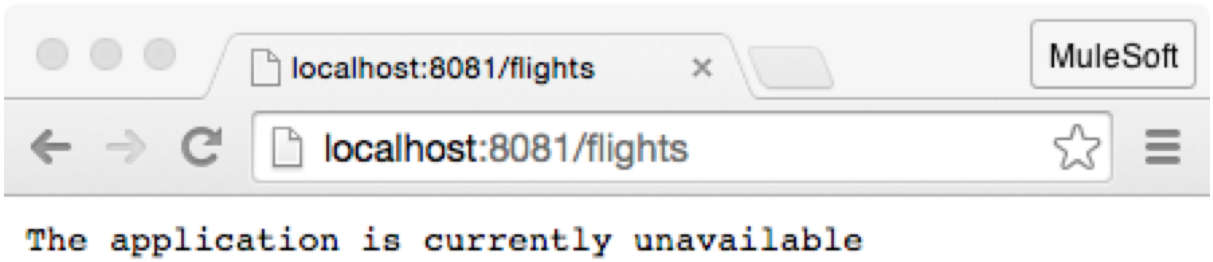
27. Save the file and run the application.
28. Make a request to http://localhost:8081/american; you should see the data unavailable error message displayed.



29. Make a request to http://localhost:8081/united; you should see the data unavailable error message displayed.



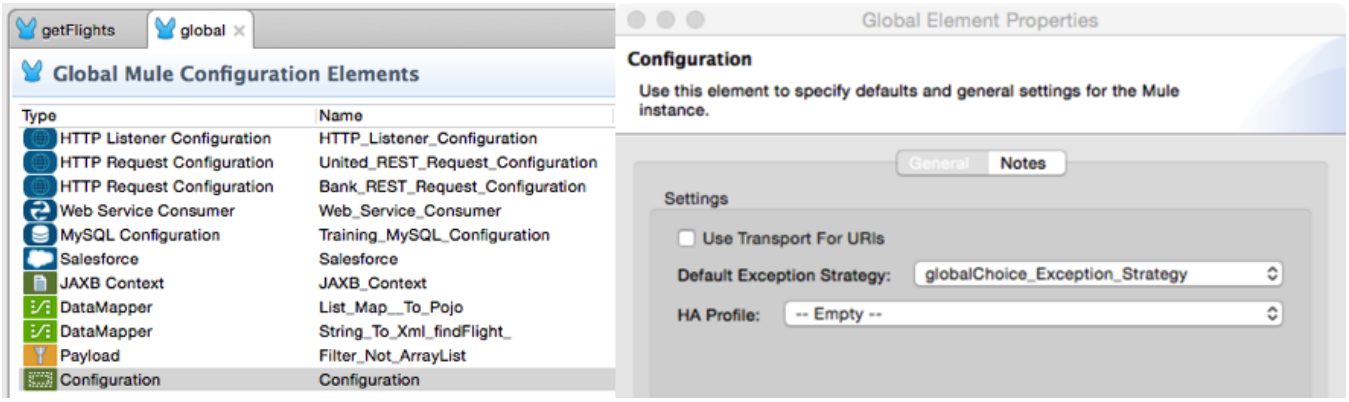
30. Make a request to http://localhost:8081/flights; you should see the application unavailable error message displayed.



Walkthrough 8-3: Specify a global default exception strategy

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

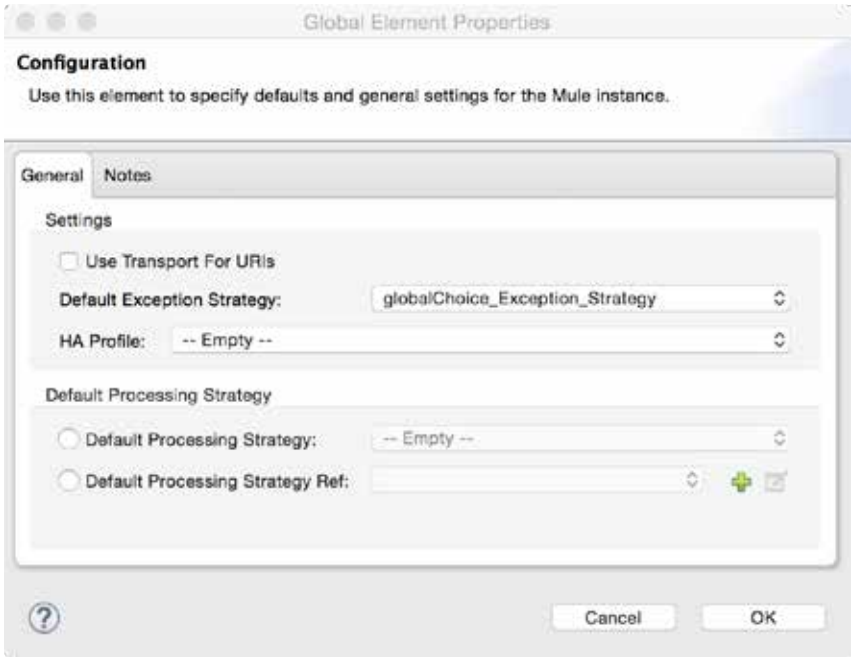
- Create a global configuration element in the global.xml file.
- Specify a default exception strategy in the global configuration element.
- Remove the existing exception handling strategies.
- Use the default exception handling strategy.



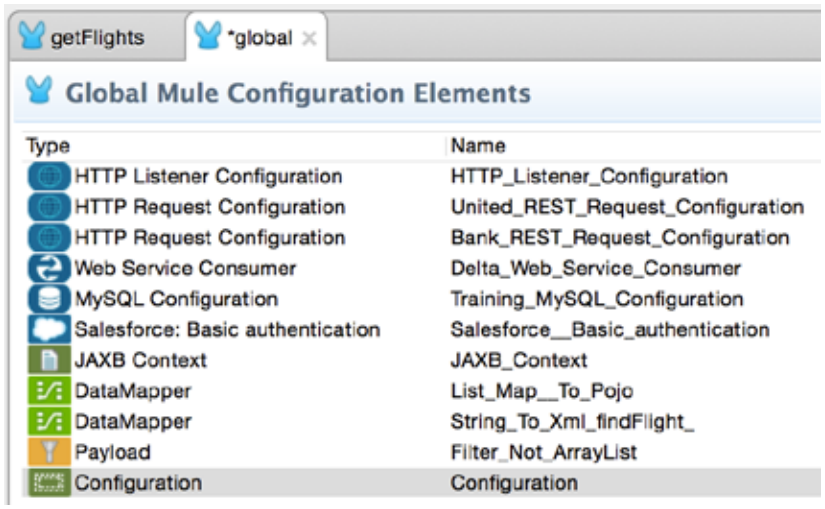
Specify a global default exception strategy

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 globalChoice_Exception_Strategy and click OK.



6. Locate your new global configuration element.

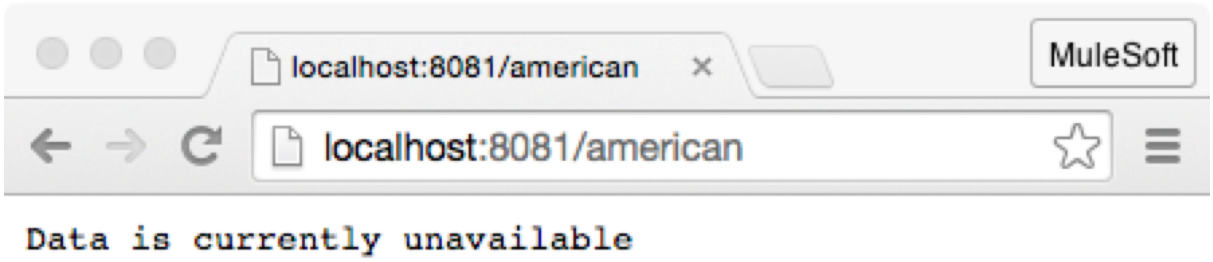


Remove the existing exception strategy references

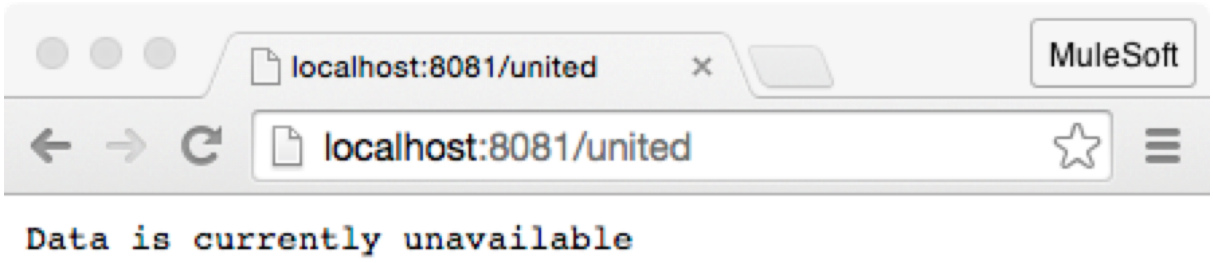
7. Return to getFlights.xml.
8. Delete the Reference Exception Strategy in getFlightsFlow.
9. Delete the Reference Exception Strategy in getFlightFormFlow.
10. Delete the Reference Exception Strategy in getAmericanFlightsFlow.
11. Delete the Reference Exception Strategy in getUnitedFlightsFlow.

Test the application

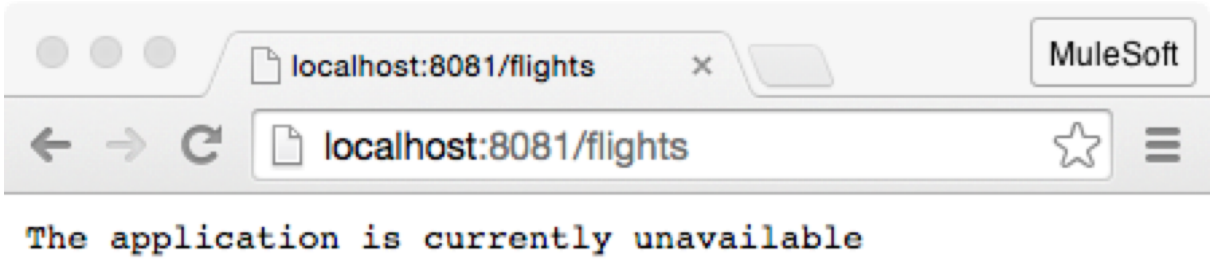
12. Save all the files and run the application.
13. Make a request to http://localhost:8081/american; you should still see the data unavailable error message displayed.



14. Make a request to http://localhost:8081/united; you should still see the data unavailable error message displayed.



15. Make a request to http://localhost:8081/flights; you should still see the application unavailable error message displayed.



Fix the application

16. In getFlightFormFlow, remove the incorrect Logger expression.
17. In getAmericanFlightsFlow, fix the database query to query the flights table.
18. In getUnitedFlightsFlow, fix the REST Request to use a path /{destination}.
19. Save the file to redeploy the application.
20. Make a request to http://localhost:8081/flights and make sure the application works as before.
21. Stop the Mule runtime.