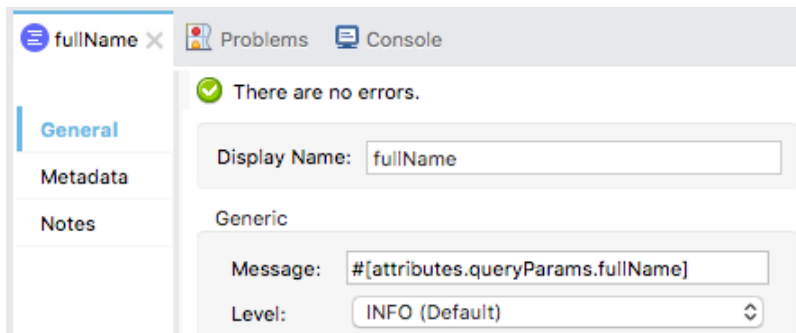# Walkthrough 6-5: Get and set event data using DataWeave expressions

In this walkthrough, you get and set event data using DataWeave expressions. You will:
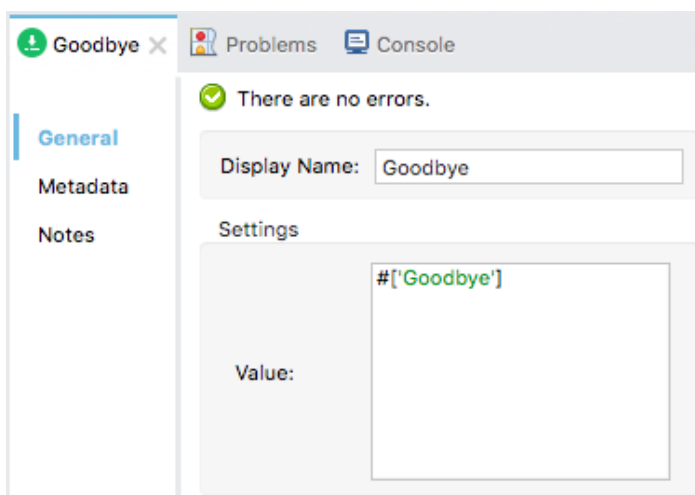
- Use expressions to set the payload and a logged value.
- Use expressions to set a response header and a request query parameter.
- In expressions, reference values for the event payload and attributes.
- Use the DataWeave upper() function and the concatenation, as, and default operators.

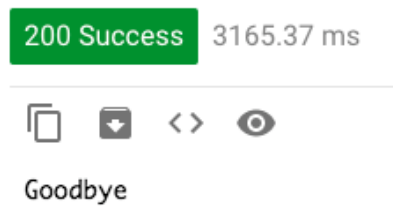

Use an expression to set the payload

1. Return to apdev-examples.xml.
2. Navigate to the properties view for the Goodbye Set Payload transformer in goodbyeFlow.
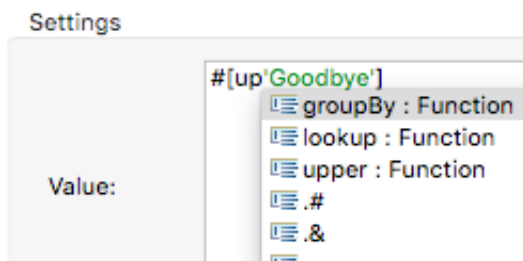3. Change the value to an expression.

```
#['Goodbye']
```


4. Run the project.

MuleSoft®

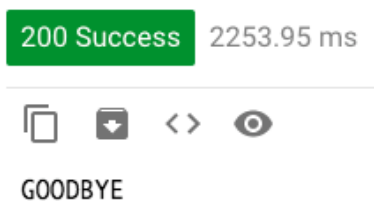5. In Advanced REST Client, send the same request; you should get the same result of Goodbye as before.

Use a DataWeave function

6. In Anypoint Studio, return to the Goodbye Set Payload properties view.
7. Inside the brackets and before the string, type the word up and press Ctrl+Spacebar.
8. In the auto-completion menu, select the upper function.

9. Add parentheses around the Goodbye string.
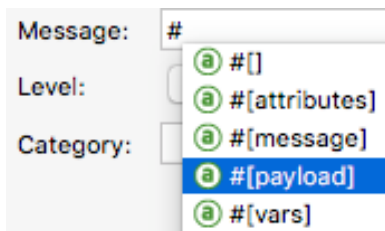
```
#[upper('Goodbye')]
```

10. Save the file to redeploy the application.
11. In Advanced REST Client, send the same request; the return string should now be in upper case.

Use an expression that references the payload in a Logger

12. Return to Anypoint Studio.
13. Navigate to the properties view for the Logger in helloFlow.
14. Change the display name to payload.

15. Type # in the message field and select #[payload] in the auto-completion menu.

16. Save the file to redeploy the application.
17. In Advanced REST Client, send the same request.
18. Return to the console in Anypoint Studio; you should see GOODBYE displayed for the second Logger instead of the entire event object.

```
INFO  2018-04-19 14:10:48,008 [[MuleRuntime].cpuLight.06: [apdev-examples].helloFlow.CPU_LITE @5dc0c3a3] [event: 0-216c8
710-4416-11e8-861e-8c85900da7e5] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: GOODBYE
```

Use a string literal and a DataWeave expression in a property value

19. Return to the properties view for the Logger in helloFlow.
20. Change the value to display the string Message in front of the payload.

```
Message: #[payload]
```

21. Save the file to redeploy the application.
22. In Advanced REST Client, send the same request.
23. Return to the console in Anypoint Studio; you should see the new string displayed.

```
INFO  2018-04-19 14:15:43,640 [[MuleRuntime].cpuLight.16: [apdev-examples].helloFlow.CPU_LITE @3dd1f9a1] [event: 0-d1a23
d00-4416-11e8-b833-8c85900da7e5] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: Message: GOODBYE
```

24. Return to the properties view for the Logger in helloFlow.

Use the DataWeave concatenation operator

25. Change the value so the string is part of the evaluated expression and use the concatenation operator.

```
#['Message: ' ++ payload]
```

26. Add \n in front of the message to display it on a new line.

```
#['\nMessage: ' ++ payload]
```

27. Save the file to redeploy the application.

28. In Advanced REST Client, send the same request.

29. Return to the console in Anypoint Studio; you should see the string displayed on a new line.

```
INFO  2018-04-19 14:18:29,925 [[MuleRuntime].cpuLight.09: [apdev-examples].helloFlow.CPU_LITE @1d096383] [event: 0-34ddd
550-4417-11e8-b833-8c85900da7e5] org.mule.runtime.core.internal.processor.LoggerMessageProcessor:
Message: GOODBYE
```
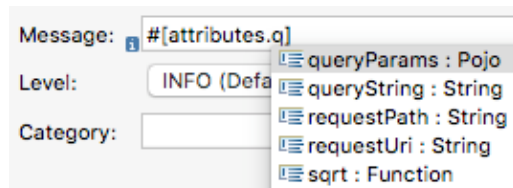
Use an expression that references an attribute in a Logger

30. Return to the properties view for the Logger in goodbyeFlow.

31. Change the display name to fullName.

32. Type # in the message field and select #[attributes] in the auto-completion menu.

33. At the end of attributes, add a period, type q and select queryParams in the auto-completion menu.


```
#[attributes.queryParams]
```

34. Click Apply Changes to redeploy the application.

35. In Advanced REST Client, send the same request.

36. Return to the Anypoint Studio console; you should see an object displayed by the first Logger.

```
INFO  2018-04-19 14:25:27,821 [[MuleRuntime].cpuLight.15: [apdev-examples].goodbyeFlow.CPU_LITE @7c601de9] [event: 0-2df
95920-4418-11e8-b833-8c85900da7e5] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: {fullName=Max Mule}
```

37. Modify the message for the Logger in goodbyeFlow to display the value of the query parameter.

```
#[attributes.queryParams.fullName]
```

38. Click Apply Changes to redeploy the application.

39. In Advanced REST Client, send the same request.

40. Return to the console; you should now see the value of the parameter displayed.

```
INFO  2018-04-19 14:26:22,192 [[MuleRuntime].cpuLight.03: [apdev-examples].goodbyeFlow.CPU_LITE @1e7dbbd3] [event: 0-4e6
55dd0-4418-11e8-b833-8c85900da7e5] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: Max Mule
```
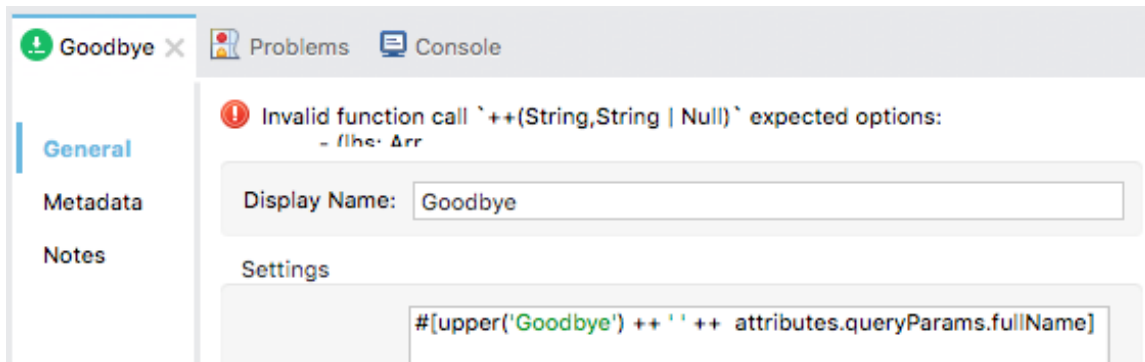
Use an expression that references an attribute when setting the payload

41. Navigate to the properties view for the Goodbye Set Payload in goodbyeFlow.

42. Use the concatenation operator to also display the value of the query parameter separated by a space.

```
#[upper('Goodbye') ++ ' ' ++ attributes.queryParams.fullName]
```
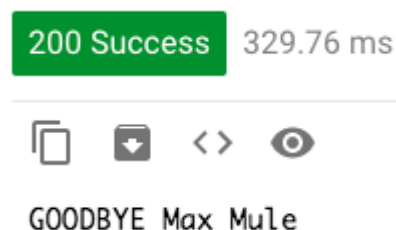
43. Review the error.

Use the as operator to coerce the attribute to a String

44. Use the as operator to also display the value of the query parameter separated by a space.

```
#[upper('Goodbye') ++ ' ' ++ attributes.queryParams.fullName as
String]
```

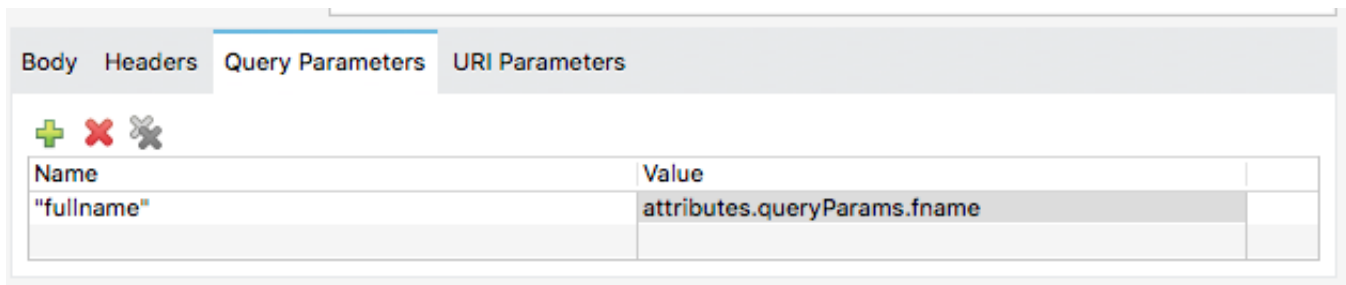45. Click Apply Changes to redeploy the application.
46. In Advanced REST Client, send the same request; you should now also see the name displayed.

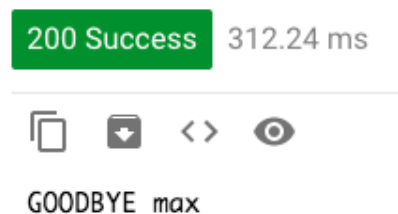## Use an expression to set a request header

47. Return to the Anypoint Studio and navigate to the properties view for the GET /goodbye HTTP Request in helloFlow.
48. In the Request section, select the Query Parameters tab.

49. Change the value of fullName to the value of the fname query parameter.

| Body | Headers | Query Parameters | URI Parameters |

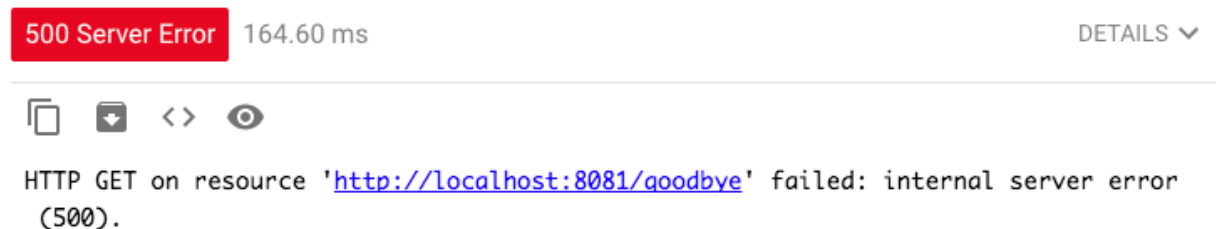| Name	Value
"fullname"	attributes.queryParams.fname

50. Click Apply Changes to redeploy the application.

51. In Advanced REST Client, send the same request; you should now see the name of the fname query parameter displayed.

200 Success 312.24 ms

GOODBYE max

Make a request and do not send a query parameter

52. Remove the query parameters and make a request to http://localhost:8081/hello; you should get an error.

500 Server Error 164.60 ms   DETAILS ∨

HTTP GET on resource 'http://localhost:8081/goodbye' failed: internal server error (500).

Set a default value in an expression

53. Return to the Anypoint Studio and navigate to the properties view for the Goodbye Set Payload in goodbyeFlow.
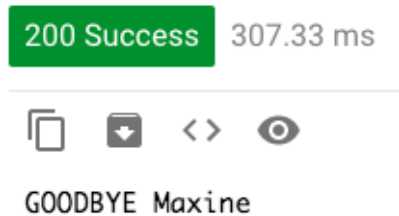
54. Remove as String.

55. Use the default operator to add a default value of Maxine.

```
#[upper('Goodbye') ++ ' ' ++ (attributes.queryParams.fullName default
'Maxine')]
```

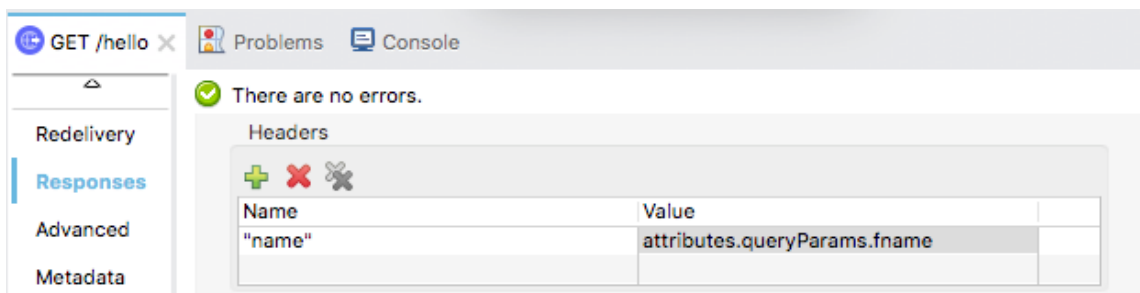56. Click Apply Changes to redeploy the application.

MuleSoft®

57. In Advanced REST Client, send the same request; you should now see the default value Maxine displayed.
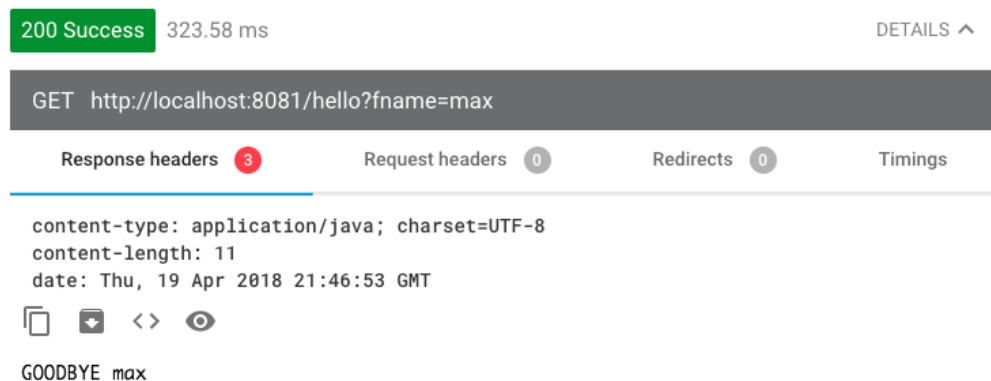

Use a query parameter in the expression for a response header

58. Return to the Anypoint Studio and navigate to the properties view for the GET /hello HTTP Listener.
59. Select the Responses tab.
60. Change the name header value to the value of the fname query parameter.

```
attributes.queryParams.fname
```


61. Click Apply Changes to redeploy the application.
62. In Advanced REST Client, add a fname and set it equal to max or some other value and send the request.
63. Look at the response headers; you should no longer see a name header.

Debug and verify the query parameter is sent with the request

64. Return to Anypoint Studio.

65. Stop and then debug the project.

66. Return to Advanced REST Client and send the same request.

67. Return to the Mule Debugger and look at the attributes and query parameters; you should see the fname query parameter.

```
ⓐ method          GET
▶ ⓔ queryParams    MultiMap{[fname=[max]]}
  ⓐ queryString   fname=max
  ⓐ relativePath  /hello
```

68. Step into goodbyeFlow and look at the attributes and query parameters; you should see the fullName query parameter.

```
ⓐ method          GET
▶ ⓔ queryParams    MultiMap{[fullName=[max]]}
  ⓐ queryString   fullName=max
  ⓐ relativePath  /goodbye
```

69. Step back to helloFlow and look at the value of the attributes; you should not see any query parameters.

```
▼ ⓔ Attributes           org.mule.extension.http.api.H
    ⓐ DOUBLE_TAB
  ▶ ⓔ headers            MultiMap{[content-type=[app
    ⓐ reasonPhrase
    ⓐ serialVersionUID   -3131769059554988414
    ⓐ statusCode         200
    ⓐ TAB
```

70. Step through the rest of the application.

71. Switch to the Mule Design perspective.