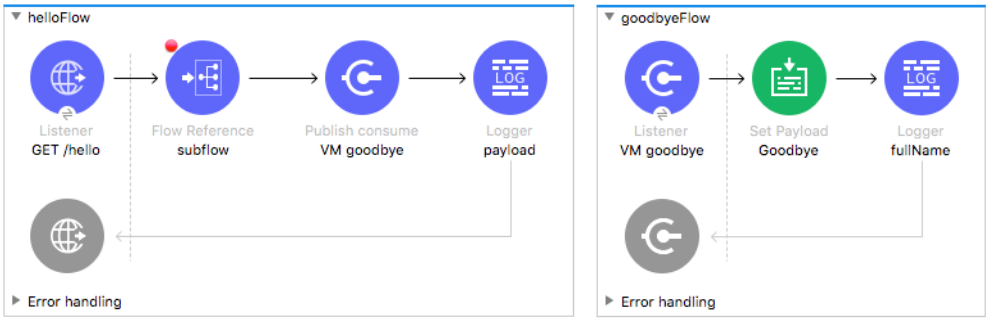
Walkthrough 7-2: Pass messages between flows using the VM connector

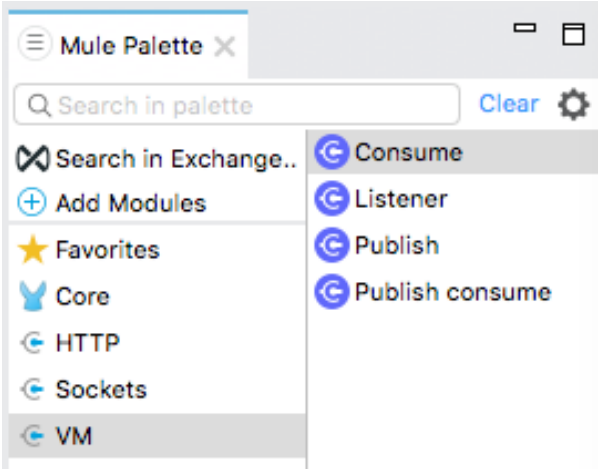
In this walkthrough, you pass messages between flows using asynchronous queues. You will:

- Pass messages between flows using the VM connector.
- Explore variable persistence with VM communication.
- Publish content to a VM queue and then wait for a response.
- Publish content to a VM queue without waiting for a response.

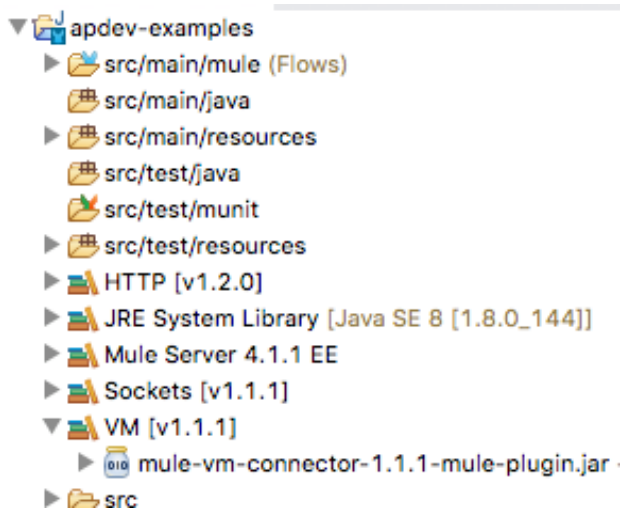


Add the VM module to the project

1. Return to apdev-examples.xml.
2. In the Mule Palette, select Add Modules.
3. Select the VM connector in the right side of the Mule Palette and drag and drop it into the left side.

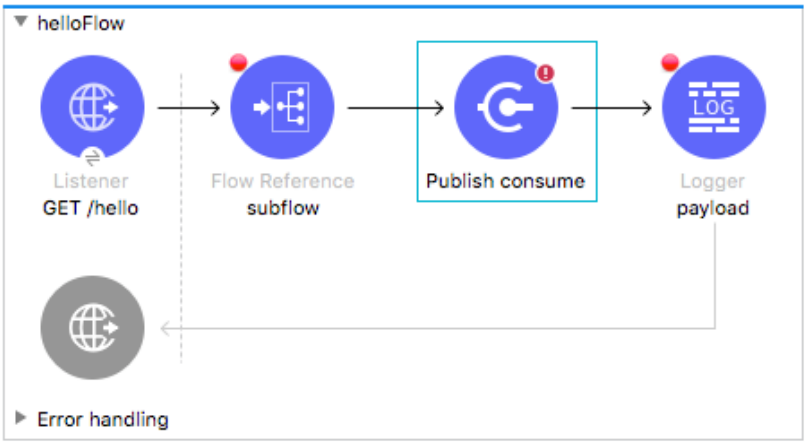


4. In the Project Explorer, locate the JAR file for the VM connector.

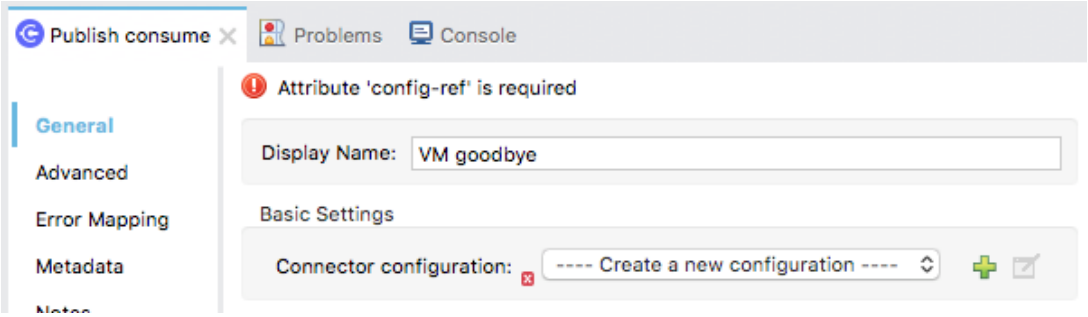


Add a VM Publish Consume operation

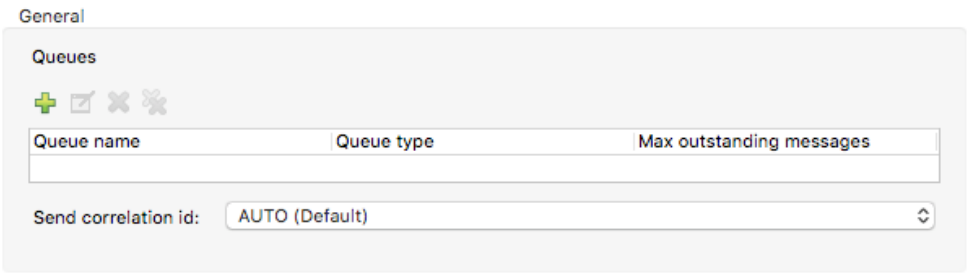
5. In helloFlow, delete the privateFlow Flow Reference.
6. Select VM in the Mule Palette.
7. Select the Publish consume operation and drag and drop it before the Logger in helloFlow.



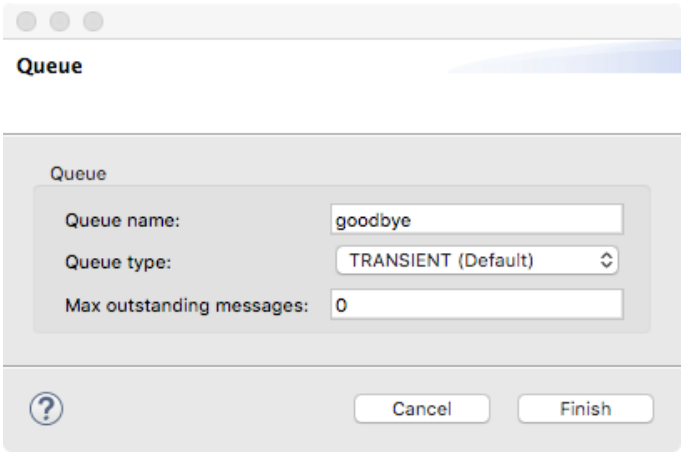
8. In the Publish consume properties view, change the display name to VM goodbye.



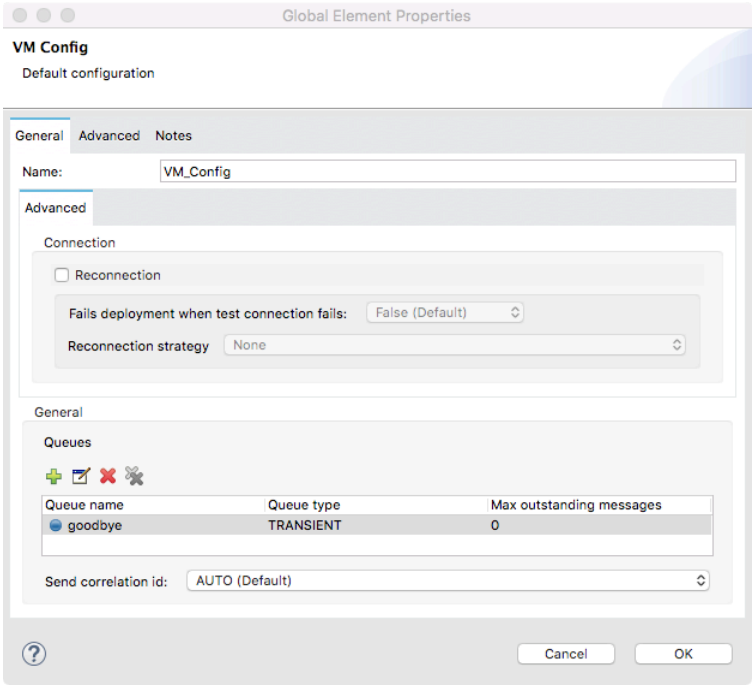
9. Click the Add button next to connector configuration.
10. In the Global Element Properties dialog box, click the Add Queue button.



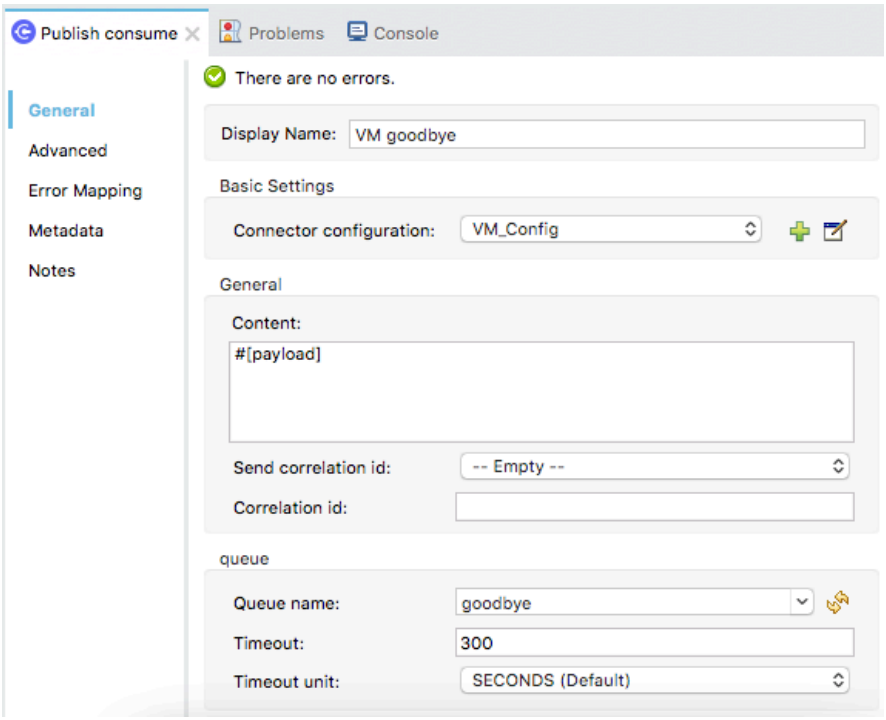
11. In the Queue dialog box, set the queue name to goodbye and click Finish.



12. In the Global Element Properties dialog box, click OK.

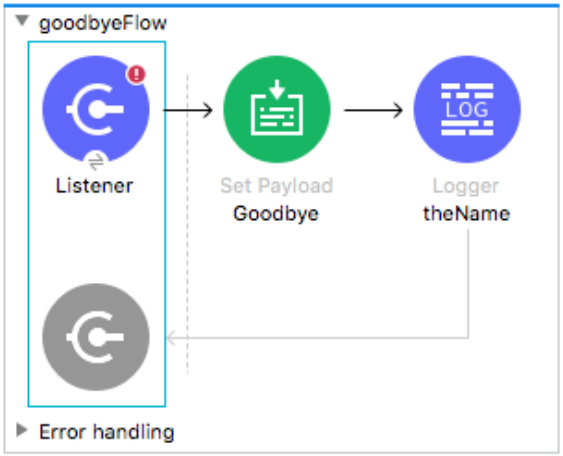


13. In the VM goodbye Publish consume properties view, set the queue name to goodbye.
14. Set the timeout to 300 seconds for debugging purposes.



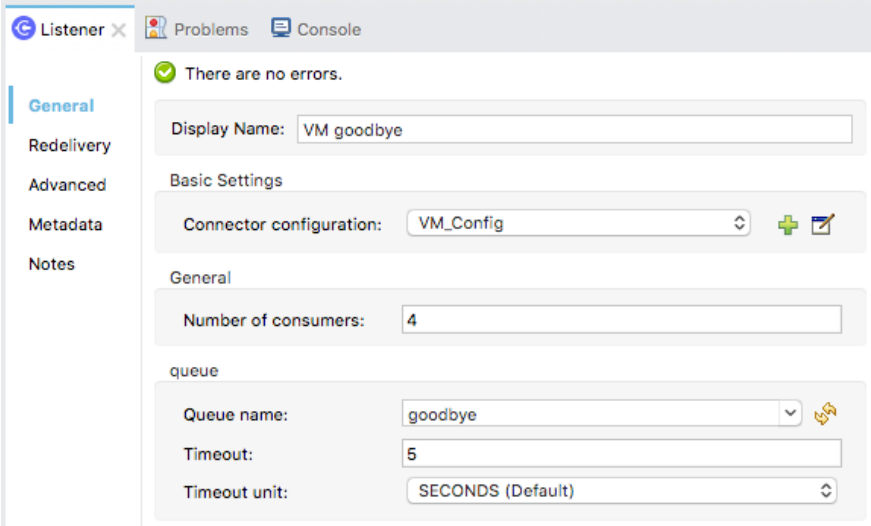
Add a VM Listener

15. Expand goodbyeFlow and delete its HTTP Listener.
16. Locate the Listener operation for the VM connector in the right side of the Mule Palette and drag and drop it in the source section of goodbyeFlow.



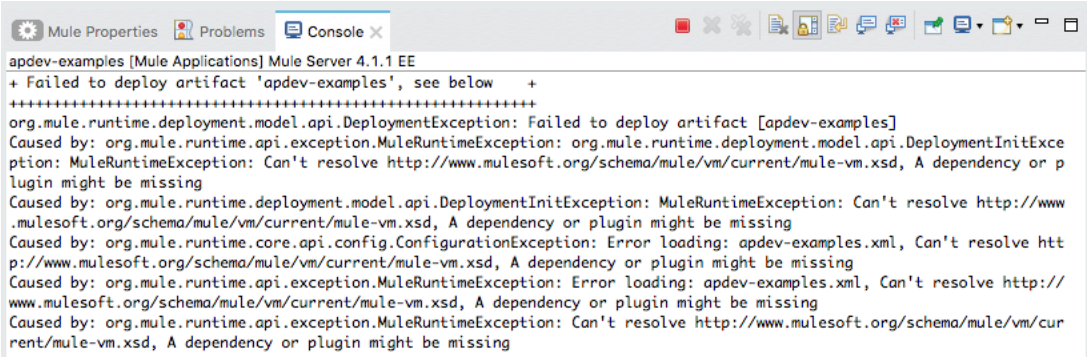
17. In the VM Listener properties view, change the display name to VM goodbye.
18. Set the connector configuration to the existing VM_Config

19. Set the queue name to goodbye.



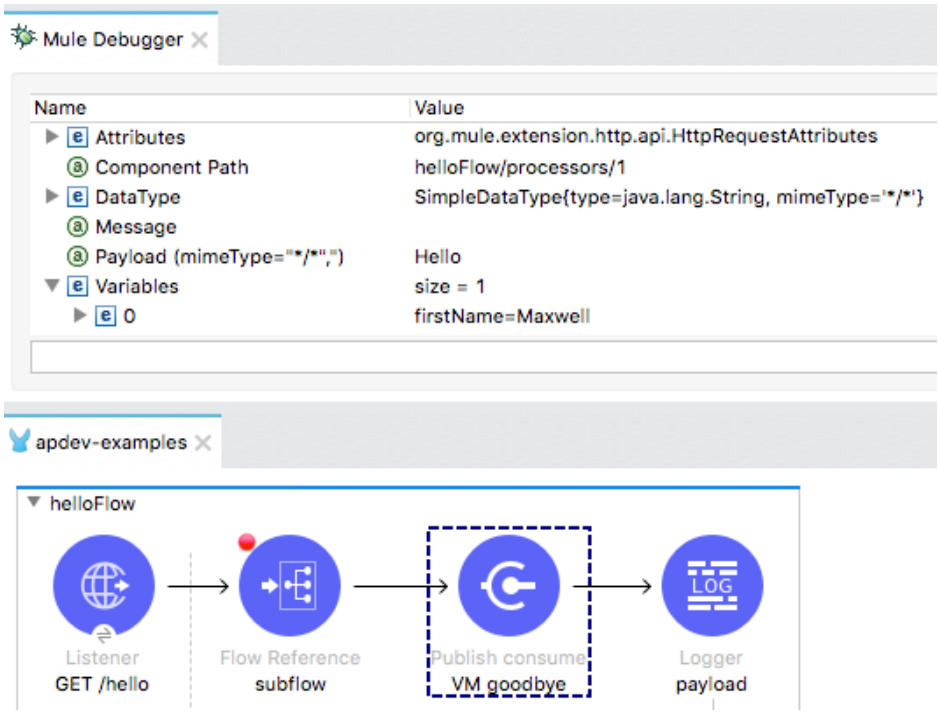
Debug the application

20. Save the file to redeploy the project in debug mode; you should get an error that a dependency is missing.

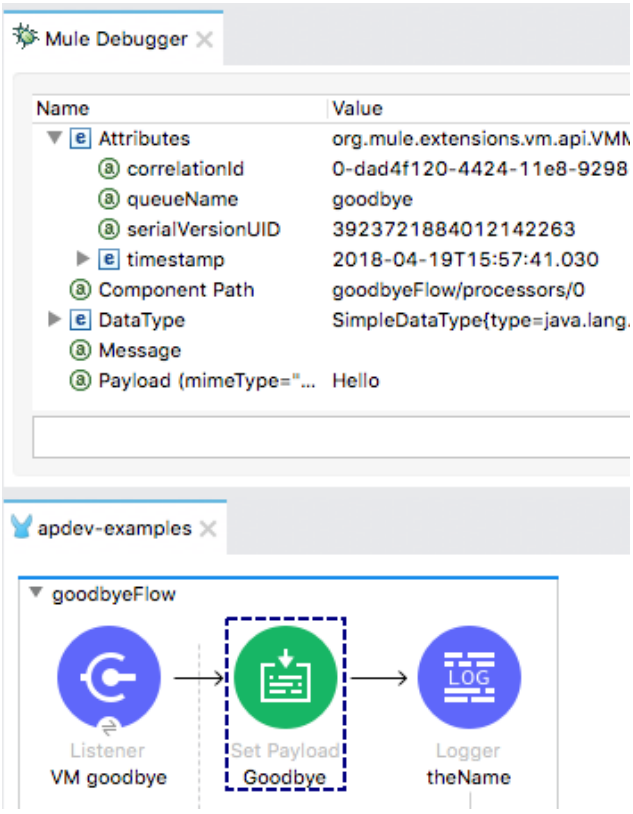


21. Stop the project.
22. Debug the project; the application should successfully deploy.
23. In Advanced REST Client, send the same request.
24. In the Mule Debugger, step through the application to the VM Publish consume.

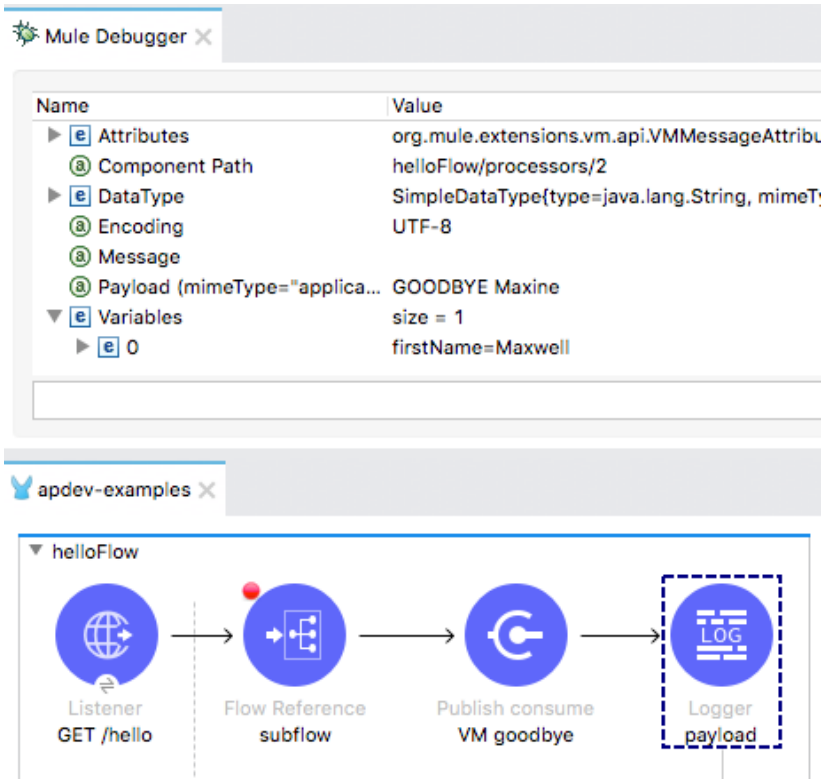
25. Look at the payload, attributes, and variables.



26. Step into goodbyeFlow.
27. Look at the payload, attributes, and variables (or lack thereof).



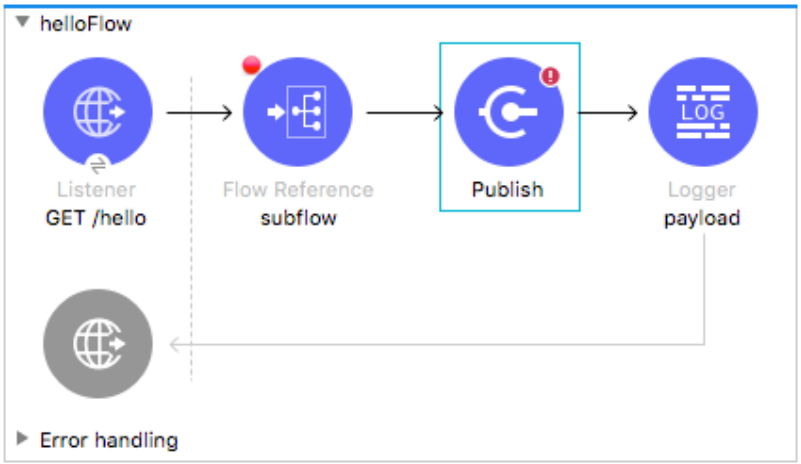
28. Step through the flow until the event returns to helloFlow.
29. Look at the payload, attributes, and variables.



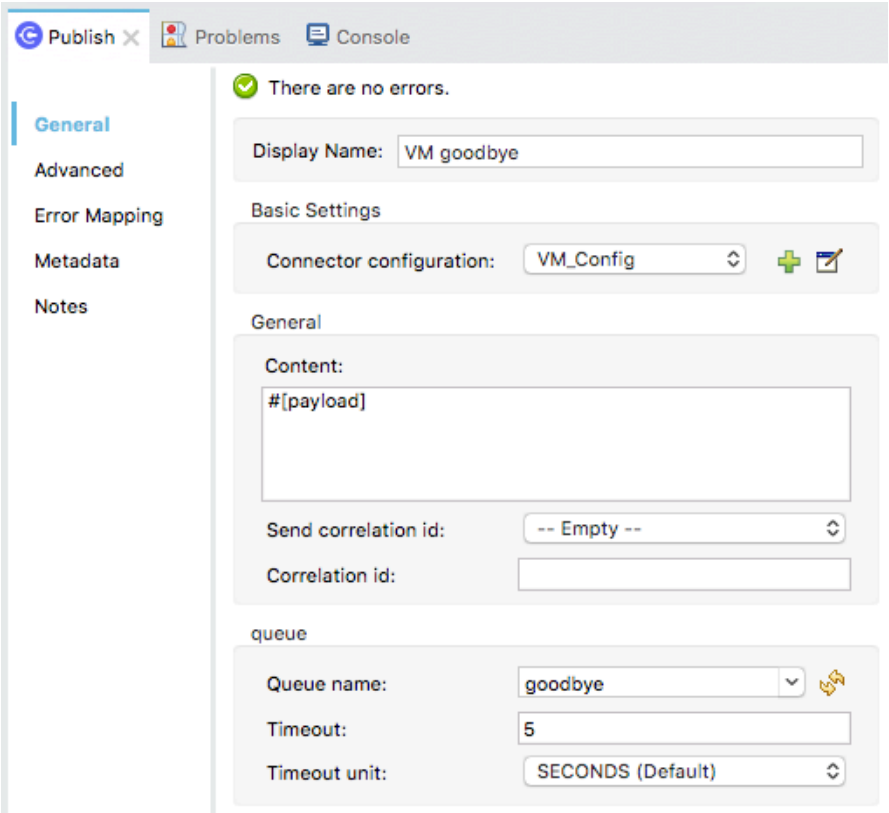
30. Step through the rest of the application.
31. Switch perspectives.

Change the VM Publish Consume operation to Publish

32. Delete the VM Publish consume operation in helloFlow.
33. Drag a VM Publish operation from the Mule Palette and drop it before the Logger.



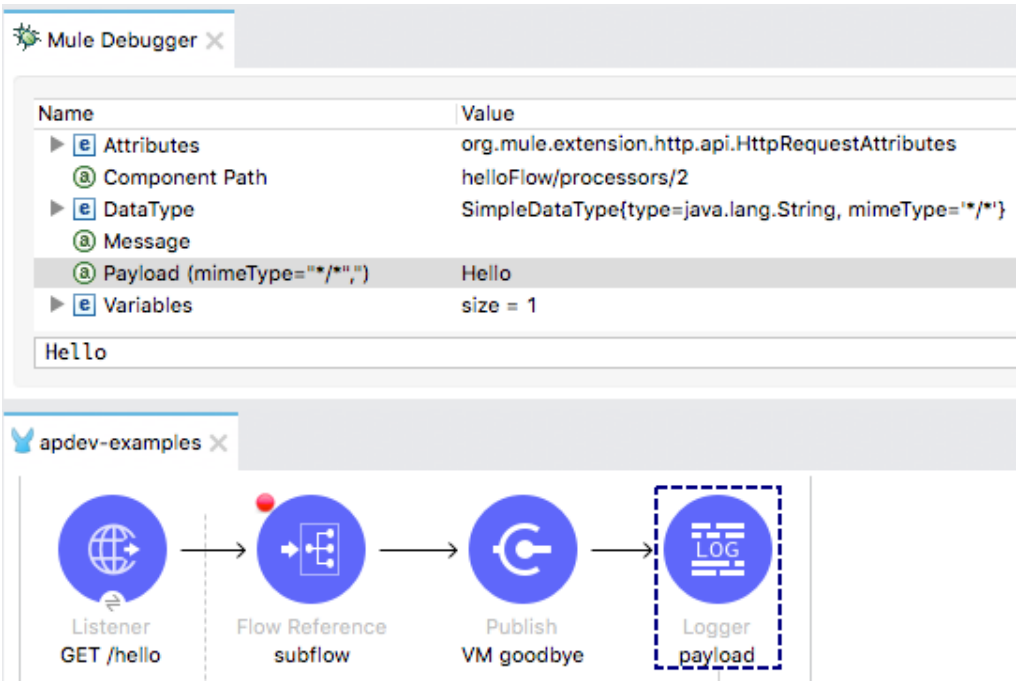
34. In the Publish properties view, set the display name to VM goodbye.
35. Set the connector configuration to the existing VM_Config.
36. Set the queue name to goodbye.



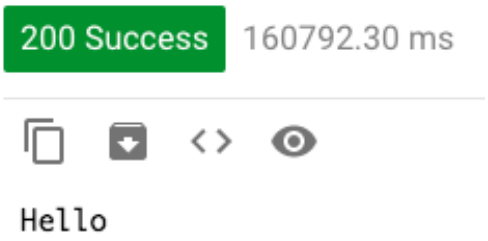
Debug the application

37. Save the file to redeploy the project in debug mode.
38. In Advanced REST Client, send the same request.
39. In the Mule Debugger, step through the application to the VM Publish operation.
40. Step again; you should step to the Logger in helloFlow.

41. Look at the value of the payload.



42. Step again; you should see execution stopped in goodbyeFlow.
43. Step to the end of the application.
44. Return to Advanced REST Client; you should see a response of Hello – not GOODBYE.



45. Return to Anypoint Studio and switch perspectives.
46. Stop the project.