Overview

●

Structuring Mule Applications

Quiz

Nice work! You got 100% correct. Please review your answers.

1.

A Mule application has two flows named parentFlow and childFlow. A variable is defined in parentFlow.

What is the scope of the variable when the parentFlow calls childFlow using a Flow Reference?

○ The variable is accessible in childFlow but is immutable

○ [✓ The variable is accessible in childFlow, can be changed, and changes are seen back in parentFlow]

○ The variable is NOT accessible in childFlow

○ The variable is accessible in childFlow, can be changed, but changes are NOT seen back in parentFlow

Correct!

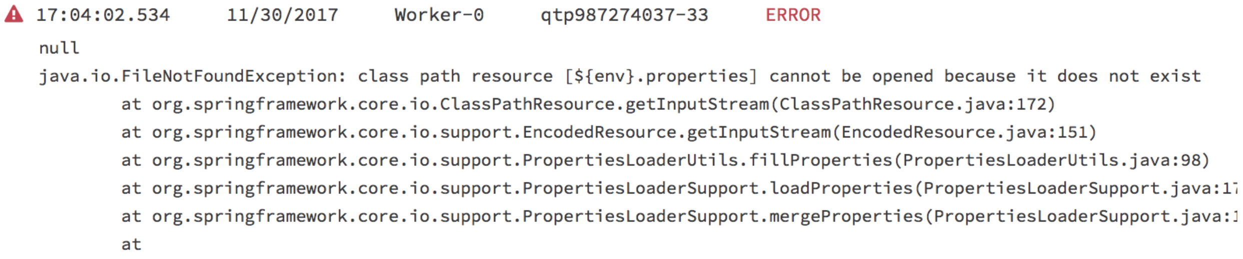
2.

Refer to the exhibit. In the deployable archive's /classes folder, there are two properties files named dev.properties and prod.properties. The Mule application fails to deploy to CloudHub through Runtime Manager with the following error message.

What could be causing this error?

DIY 7-1: Track data through a Mule application ➡

⬅ **Structuring Mule Applications**



```
⚠ 17:04:02.534     11/30/2017     Worker-0     qtp987274037-33     ERROR
  null
  java.io.FileNotFoundException: class path resource [${env}.properties] cannot be opened because it does not exist
          at org.springframework.core.io.ClassPathResource.getInputStream(ClassPathResource.java:172)
          at org.springframework.core.io.support.EncodedResource.getInputStream(EncodedResource.java:151)
          at org.springframework.core.io.support.PropertiesLoaderUtils.fillProperties(PropertiesLoaderUtils.java:98)
          at org.springframework.core.io.support.PropertiesLoaderSupport.loadProperties(PropertiesLoaderSupport.java:17
          at org.springframework.core.io.support.PropertiesLoaderSupport.mergeProperties(PropertiesLoaderSupport.java:1
          at
```

○ The property files were NOT saved at the root level of the deployable archive

[✓ ○ The env property is NOT set in the Runtime Manager in the Mule application's Properties tab]

○ A file named ${env}.properties is NOT included in the deployable archive file

○ The dev.properties and prod.properties files were NOT uploaded to Runtime Manager

Correct!

3.

What reserved property can be defined and used in a Mule application to allow an HTTPS Listener to be accessed by external web clients after the Mule application is deployed to CloudHub?

○ ${ssl.port}

○ ${ssl.listener.port}

○ ${https.listener.port}

[✓ ○ ${https.port}]

Correct!

4.

Why must a Mule application's deployable archive package all its dependencies in order to be deployed to CloudHub?

○ The online logging service requires access to ALL project dependencies to log the appropriate Java classes used in the Mule application

[✓ ○ CloudHub workers CANNOT download ALL possible project dependencies a project may contain]

○ CloudHub workers needs to compare the current dependencies with the LATEST project dependencies from the MuleSoft repository

○ MuleSoft support requires access to ALL project dependencies for future online troubleshooting

Correct!

5.

In what file does the Mule project keep track of all of its dependencies?

○ mule-artifact.json

○ [✓ pom.xml]

○ mule-app.properties

○ global.xml

Correct!

6.

What can ONLY be done with VM connectors, and NOT with Flow References, in a single Mule application?

○ Preserve variables as the Mule event gets passed to another flow

○ Allow a flow to pass events to another flow synchronously

○ Preserve the original payload when the VM connector returns a response from a flow

○ [✓ Allow a flow to pass events to another flow asynchronously]

Correct!

Nice work! You got 100% correct. Please scroll up to review your answers.