

# MCD-LEVEL1<sup>Q&As</sup>

MuleSoft Certified Developer - Level 1 (Mule 4)

## Pass Mulesoft MCD-LEVEL1 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass2lead.com/mcd-level1.html>

100% Passing Guarantee  
100% Money Back Assurance

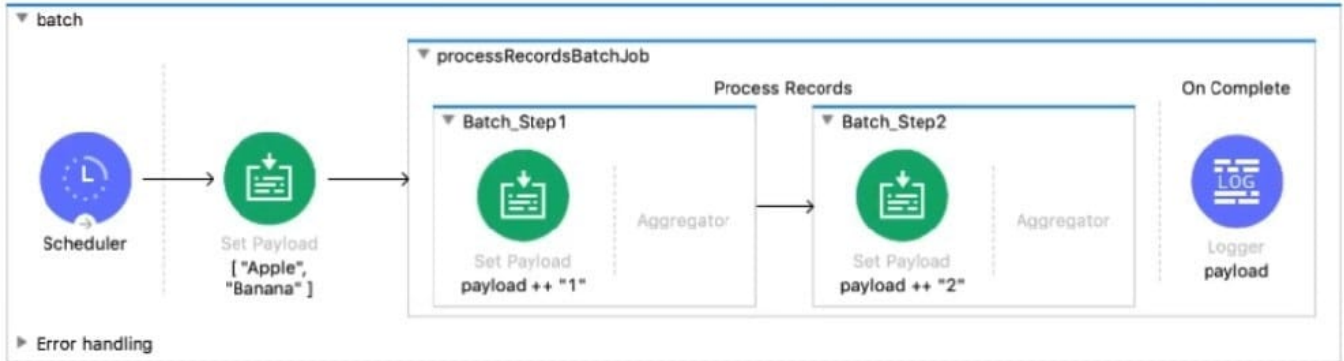
Following Questions and Answers are all new published by Mulesoft  
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



**QUESTION 1**

Refer to the exhibit. The input array of strings is passed to the batch job, which does NOT do any filtering or aggregating. What payload is logged by the Logger component?



- A. Summary report of processed records
- B. ["Apple", "Banana"]
- C. ["Apptel2", "Bananal2"]
- D. ["Apptel", "Bananal", 2]

Correct Answer: A

**QUESTION 2**

In an application network. If the implementation but not the interface of a product API changes, what needs to be done to the other APIs that consume the product API?

- A. The applications associated with the other APIs must be restarted
- B. The applications associated with the other APIs must be recoded
- C. The other APIs must be updated to consume the updated product API
- D. Nothing needs to be changed in the other APIs or their associated applications

Correct Answer: D

Correct answer is Nothing needs to be changed in the other APIs or their associated applications This is the benefit of having separate interface layer. As there are no changes to interface , no changes are required on the API\\'s which consumes this API in context

**QUESTION 3**

Mule application contains ActiveMQ JMS dependency. Mule application was compiled and run successfully in Anypoint

Studio. The mule application must now be exported from Anypoint Studio and shared with other developer. What export options should be selected to create the smallest JAR file that can be imported into other developer's Anypoint Studio and run successfully?

- A. Select only Attach Project Sources only
- B. Select both Attach Project Sources and Include project modules and dependencies option
- C. Select the Include project modules and dependencies option only
- D. De-select both Attach Project Sources and Include project modules and dependencies option

Correct Answer: A

Correct answer is Select only Attach Project Sources only You must keep the Attach Project Sources option selected to be able to import the packaged JAR file back into a Studio workspace. Deselect the Include project modules and dependencies option. This option skips bundling the actual modules and external dependencies required to run the Mule application in a Mule runtime engine, creating a lightweight JAR file package that does not include any dependencies specified in the Mule application's pom.xml file. The generated JAR file is not a functional deployable archive and cannot be deployed to a Mule runtime engine, but instead offers a way to archive only the source files that make up the Mule application. This is the same as using the `-lightWeightPackage` flag when packaging using the Mule Maven Plugin and is useful if you want to distribute your project to other Studio installations because it only keeps a reference to all its dependencies. When you import a lightweight package into Studio, all your dependencies are automatically downloaded  
MuleSoft Doc Ref : <https://docs.mulesoft.com/studio/7.8/import-export-packages>

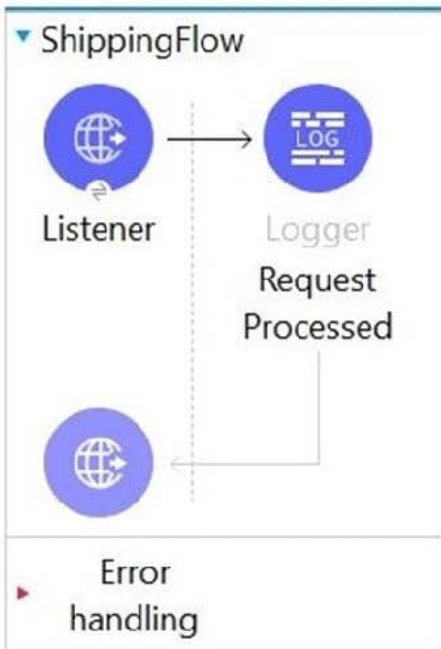
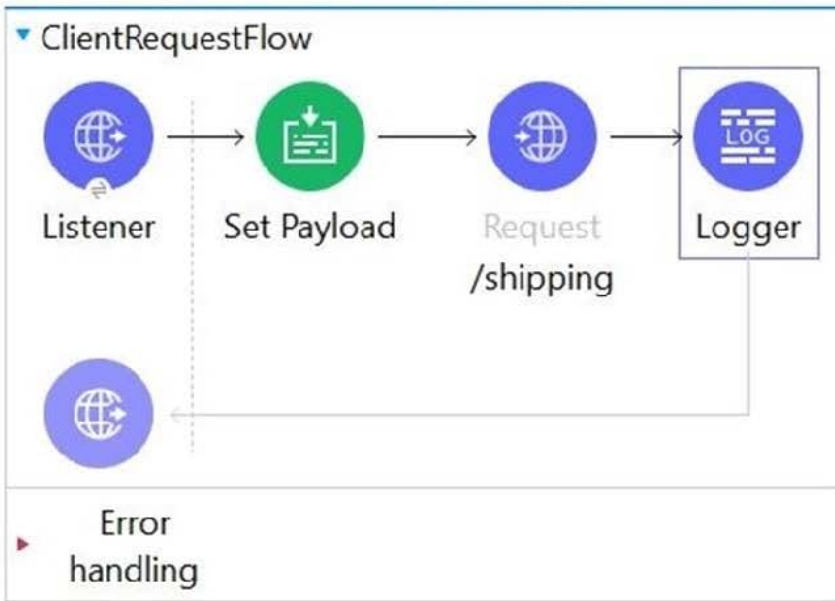
---

#### QUESTION 4

Refer to the exhibits. Client sends the request to ClientRequestFlow which calls ShippingFlow using HTTP Request activity.

During E2E testing it is found that that HTTP:METHOD\_NOT\_ALLOWED error is thrown whenever client sends request to this flow.

What attribute you would change in ClientRequestFlow to make this implementation work successfully?



```

<http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener config" doc:id="a146bd76-e3c2-4db0-95bf-c124e3ddd4d3" >
  <http:listener-connection host="0.0.0.0" port="8081" />
</http:listener-config>
<http:request-config name="HTTP_Request_configuration" doc:name="HTTP Request configuration" doc:id="953ca3a1-728e-4469-88ba-3a00e0d40a19" >
  <http:request-connection host="localhost" port="8081" />
</http:request-config>
<flow name="ClientRequestFlow" doc:id="2fde32c2-877b-4f70-b039-b6f111b3fb0f" >
  <http:listener doc:name="Client HTTP Listener" doc:id="8d3a6e62-4b26-4ec9-9f03-aa0e1d254633" config-ref="HTTP_Listener_config" path="/api/ship"/>
  <set-payload value="#{ product:'Chair', address:'Mumbai'}" doc:name="Set Payload" doc:id="03b6de3a-0a4b-4fdb-b770-f533f3ce63a0" />
  <http:request method="PUT" doc:name="/shipping" doc:id="3a6171a9-2a76-4588-bcdc-4a31ba4881b2" config-ref="HTTP_Request_configuration" path="/shipping"/>
  <logger level="INFO" doc:name="Logger" doc:id="64da1a92-a2c2-45aa-adac-c8bb17eae93b" />
</flow>
<flow name="ShippingFlow" doc:id="904402e3-c1d3-49f0-a7ef-00aa0c4c4936" >
  <http:listener doc:name="Listener" doc:id="405a1626-11b1-41f2-828d-c4d3bcd64b5" config-ref="HTTP_Listener_config" path="/shipping" allowedMethods="POST"/>
  <logger level="INFO" doc:name="Request Processed" doc:id="b7acd0c2-1621-4853-a4be-558b96412650" message="Request Processed"/>
</flow>
  
```

```

*****
Message           : HTTP PUT on resource 'http://localhost:8081/shipping' failed: method not allowed (405).
Element           : ClientRequestFlow/processors/1 @ mule_app:flow1.xml:68 (/shipping)
Element DSL       : <http:request method="PUT" doc:name="/shipping" doc:id="3a6171a9-2a76-4588-bcdc-4a31ba4881b2" config-
ref="HTTP_Request_configuration" path="/shipping"></http:request>
Error type        : HTTP:METHOD_NOT_ALLOWED
FlowStack         : at ClientRequestFlow(ClientRequestFlow/processors/1 @ mule_app:flow1.xml:68 (/shipping))

(set debug level logging or '-Dmule.verbose.exceptions=true' for everything)
*****
  
```

- A. Change the method attribute value to "\*"\\\"'
- B. Change the path attribute value to "/api/ship"
- C. Change the allowed method attributes value to "POST"
- D. Change the protocol attribute value to "HTTPS"

Correct Answer: C

Correct answer is Change the method attributes value to "POST".

It can be fixed in either of the two ways as below.

- 1) Changing method attribute to POST in ClientRequestFlow
  - 2) Setting allowedMethods as PUT in ShippingFlow (but doesn't fit as question mentions about changing ClientRequestFlow)
- 

#### QUESTION 5

A shopping API contains a method to look up store details by department.

To get the information for a particular store, web clients will submit requests with a query parameter named department and uri parameter named storeId

What is valid RAML snippet that supports requests from a web client to get a data for a specific storeId and department name?

- A. 1./department:  
2.  
get:  
3.  
uriParameter:  
4.  
storeId:
- B. 1.get: 2.uriParameter: 3.{storeId}: 4.queryParameter: 5.department:
- C. 1.get:  
2.  
queryParameter:  
3.  
department:

4.

uriParameter:

5.

{storeId}:

D. 1./{storeId}:

2.

get:

3.

queryParameter:

4.

department:

Correct Answer: D

[MCD-LEVEL1 PDF Dumps](#)

[MCD-LEVEL1 VCE Dumps](#)

[MCD-LEVEL1 Practice Test](#)