



# 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.geekcert.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.

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            example: |
              [
                {
                  "plan_type": "Super Saver 500",
                  "plan_details": "all-inclusive",
                  "monthly_discount": 0.10
                },
                {
                  "plan_type": "Business Plus 1000",
                  "plan_details": "business package",
                  "monthly_discount": 0.20
                }
              ]
```

The API needs to be updated using the company-wide standard for the Plan data type. The Object data type has already been published in Anypoint Exchange with the global reference . ACME/DataTypes/PlanDataType.raml. What is a valid RAML specification that reuses the Plan data type?



A. )

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

dataTypes:
  Plan: !include ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !include ACME/Examples/PlanExamples.raml
```

B. )

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

types:
  Plan: !reference ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !reference ACME/Examples/PlanExamples.raml
```

C. )

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

dataTypes:
  Plan: !reference ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !reference ACME/Examples/PlanExamples.raml
```

D. )

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

types:
  Plan: !include ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !include ACME/Examples/PlanExamples.raml
```



- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: D

As can be seen in RAML, POST expects input in application/json format which eliminates two of the options as two options are in xml format. Now out of the two remaining options, one has id field in request which is only mentioned for get

response and not for POST request. Hence id field is not expected in POST request.

Hence correct answer is

```
{  
  "name": "GoerdiLa Forge",  
  "address": "1 Westland CA",  
  "customer_since": "2014-01-04",  
  "balance": "4829.29",  
  "bank_agend_id": "12556"  
}
```

---

## QUESTION 2

Which one of them is NOT a flow in Mule?

- A. sync flow
- B. subflow
- C. async sub flow
- D. async flow

Correct Answer: C

Correct answer is async sub flow. Rest are valid flows in Mule. Sub flow is always synchronous.

---

## QUESTION 3

Refer to the exhibit.



```
##RAML 1.0
title: ACME Insurance API
```

```
/users:
  post:
    headers:
      username: string
      password: string
    body:
      application/json:
```

What is the correct way to create a user?

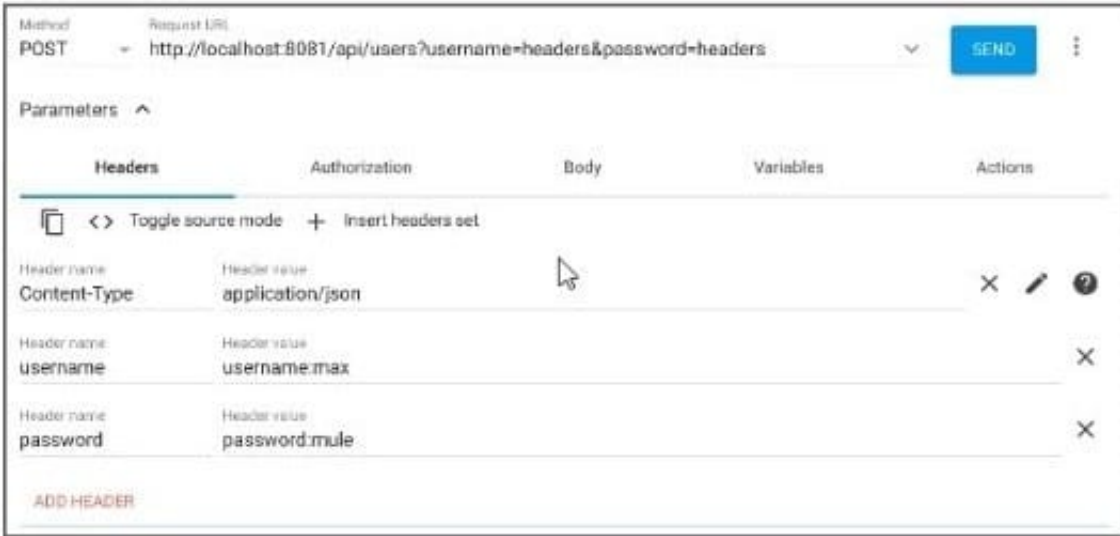


A.

B.

C.



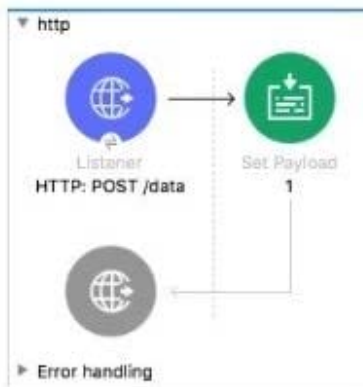
D. 

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: B

#### QUESTION 4

Refer to the exhibit.



```
<flow name="main">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <http:request method="POST" doc:name="HTTP: POST /data" url="http://localhost:8081/data"/>
  <jms:publish-consume doc:name="JMS: num1" config-ref="JMS_Config" destination="num1"/>
  <jms:publish doc:name="JMS: num2" config-ref="JMS_Config" destination="num2"/>
  <set-payload value="#[payload + 1]" doc:name="payload + 1" />
</flow>
```

What payload is returned from a request to http://localhost.8081/

Refer to the exhibits, what payload is returned from a request to http://localhost;8081/?

- A. 1
- B. 2
- C. 3
- D. 4

Correct Answer: C

The flow can be described as below. 1) First HTTP POST requests is made in which payload is set to 1 and it gets returned to our main flow. 2) Second call is initiated for JMS Publish Consume JMS: num1 which add 1 to the payload which makes it as 2. Note that publish consume is a synchronous operation. Hence payload is returned to main flow. 3) Third call is initiated for JMS Publish JMS: num2 which add 1 to the payload. Note that publish is asynchronous operation. Hence payload is never returned to main flow. So payload in main flow is still 2. 4) Finally Set Payload increments payload by 1 making payload as 3 which is returned by the flow. Hence option C is the correct answer.



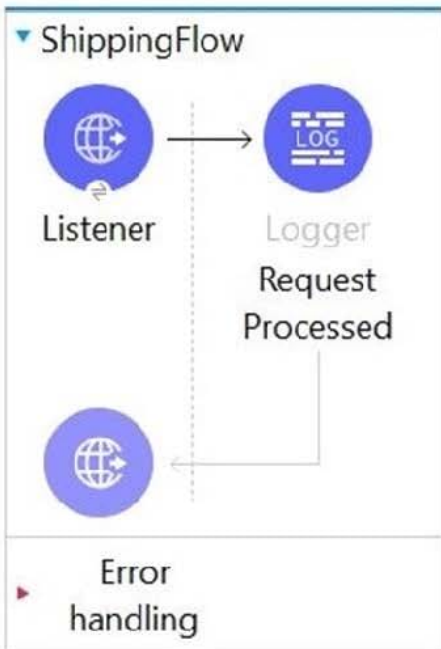
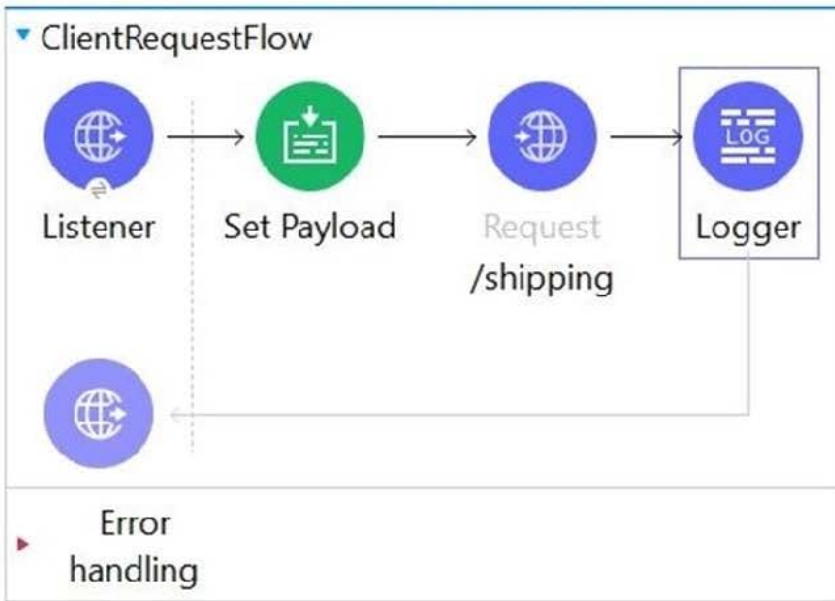


### QUESTION 5

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-3a0e0d40a19" >
  <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)

[Latest MCD-LEVEL1 Dumps](#)

[MCD-LEVEL1 Practice Test](#)

[MCD-LEVEL1 Brindumps](#)