



# 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.pass4itsure.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**

An API implementation has been deployed to CloudHub and now needs to be governed. IT will not allocate additional vCore for a new Mule application to act as an API proxy.

What is the next step to preserve the current vCore usage, but still allow the Mule application to be managed by API Manager?

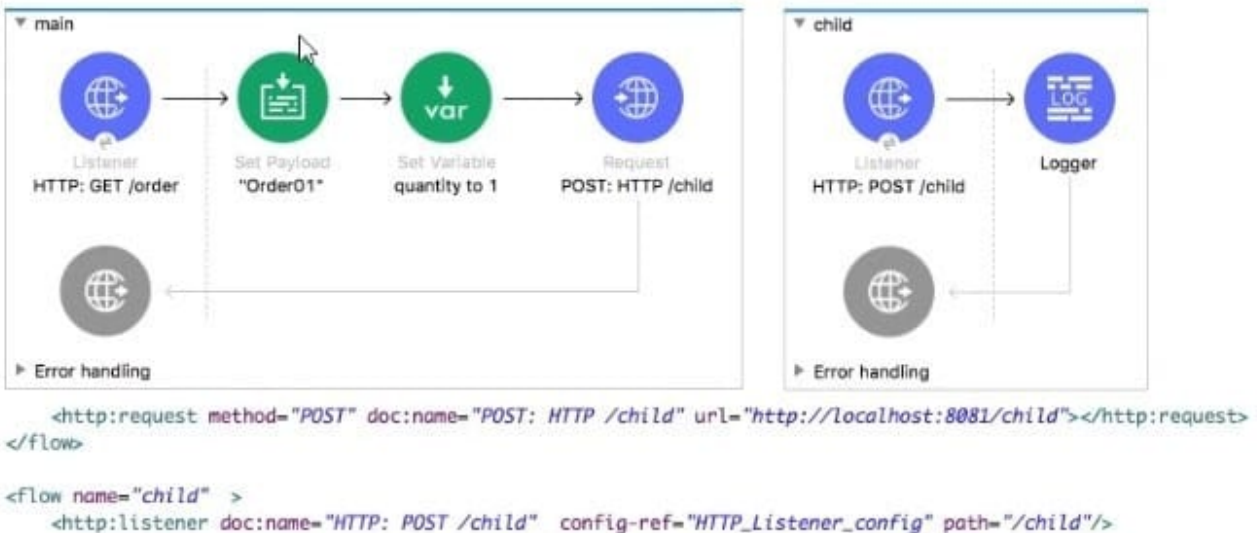
- A. Register the same API implementation in Runtime Manager to connect to API Manager
- B. Modify the API implementation to use auto-discovery to register with API Manager
- C. Upload the Mule application's JAR file to the API instance in API Manager
- D. Deploy the same API implementation behind a VPC and configure the VPC to connect to API Manager

Correct Answer: B

Correct answer is Modify the API implementation to use auto-discovery to register with API Manager API Autodiscovery Configuring autodiscovery allows a deployed Mule runtime engine (Mule) application to connect with API Manager to download and manage policies and to generate analytics data. Additionally, with autodiscovery, you can configure your Mule applications to act as their own API proxy. When autodiscovery is correctly configured in your Mule application, you can say that your application's API is tracked by (green dot) or paired to API Manager. You can associate an API in a Mule setup with only one autodiscovery instance at a given time. MuleSoft Doc Ref : <https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept>

**QUESTION 2**

Refer to the exhibits.



The main flow contains an HTTP Request. The HTTP Listeners and HTTP Request use default configurations.

What values are accessible in the child flow after a web client submits a request to <http://localhost:8081/order>? col or = red?



- A. payload
- B. payload quantity var
- C. payload color query param
- D. payload quantity var color query param

Correct Answer: A

---

### QUESTION 3

What DataWeave expression transforms the example XML input to the CSV output?



```
<?xml version="1.0" encoding="UTF-8"?>
<sale>
  <item itemId="592" saleId="1000">
    <shipping>international</shipping>
    <desc>T-shirt Navy</desc>
    <size>L</size>
    <quantity>1</quantity>
    <price>20</price>
  </item>
  <item itemId="972" saleId="1000">
    <shipping>domestic</shipping>
    <desc>Cargo Shorts</desc>
    <size>XL</size>
    <quantity>2</quantity>
    <price>30</price>
  </item>
</sale>
```

```
Output Payload • Preview
1 %dw 2.0
2 output application/csv
3 ---
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
index,sale,itemName,itemPrice,item
0,1000,T-shirt Navy,20,592
1,1000,Cargo Shorts,60,972
```

- A. `payload.sale.*item map ( (value,index) -> {  
 index: index,  
 sale: value.@saleId,  
 itemName: value.desc,  
 itemPrice: (value.price) * (value.quantity),  
 item: value.@itemId  
} )`
- B. `payload.sale.*item map ( (value,index) -> {  
 index: index,  
 sale: value.saleId,  
 itemName: value.desc,  
 itemPrice: (value.price) * (value.quantity),  
 item: value.itemId  
} )`
- C. `payload.sale.item map ( (value,index) -> {  
 index: index,  
 sale: value.@saleId,  
 itemName: value.desc,  
 itemPrice: (value.price) * (value.quantity),  
 item: value.@itemId  
} )`
- D. `payload.sale.item map ( (value,index) -> {  
 index: index,  
 sale: value.saleId,  
 itemName: value.desc,  
 itemPrice: (value.price) * (value.quantity),  
 item: value.itemId  
} )`



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

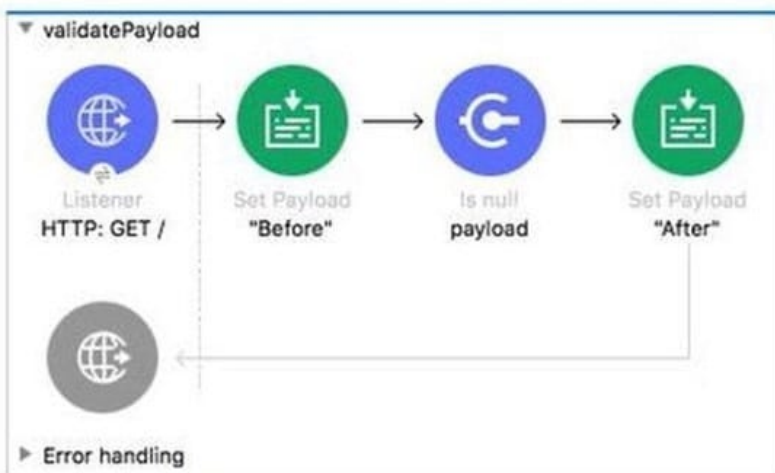
Correct Answer: A

Correct answer is as below. Attributes in the incoming xml payload are always accessed using @. Similarly \*item is required as we have multiple items in the request %dw 2.0 output application/csv

```
payload.sale.*item map ((value, index) -> {  
  index: index,  
  sale: value.@saleId,  
  itemName: value.desc,  
  itemPrice: (value.quantity) * (value.price),  
  item: value.@itemId  
})
```

#### QUESTION 4

Refer to the exhibits.



```
<flow name="validatePayload" >  
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />  
  <set-payload value="Before" doc:name="Before" />  
  <validation:is-null doc:name="payload" value="#[payload]" message="Validation Error"/>  
  <set-payload value="After" doc:name="After" />  
</flow>
```



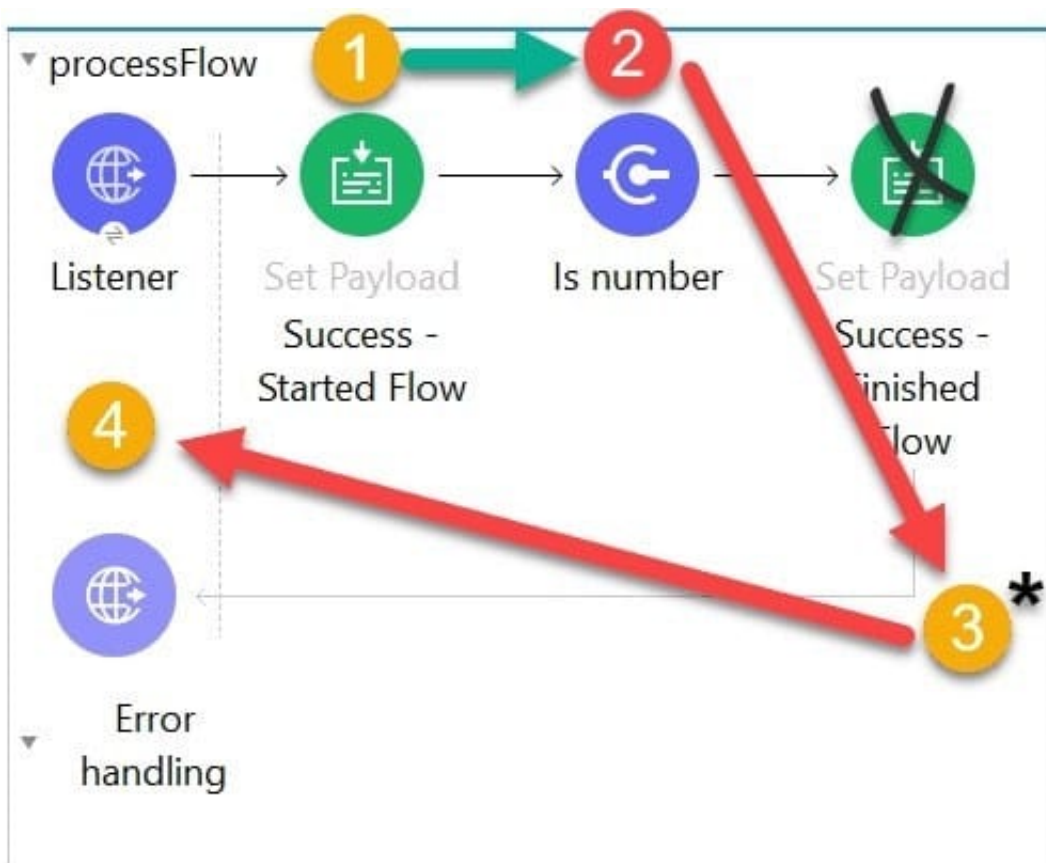
What is the response when a client submits a request to `http://localhost:8081`?

- A. After
- B. null
- C. Before
- D. Validation error

Correct Answer: D

Here's specifically what is happening here:

- 1) Payload is successfully set to "Before"
- 2) Is null validation is used which will pass the message only if payload is null. In this case as payload is not null, it creates an Error Object. Flow execution stops #[error.description] = "Validation error"
- 3) Because no error handler is defined, the Mule default error handler handles the error
- 4) "Validation error" is the error message returned to the requestor in the body of the HTTP request with HTTP Status Code: 500 Reference diagram:



#### QUESTION 5

A function named `newProdCode` needs to be defined that accepts two input parameters, an integer value for `itemID` and



a string value for productCategory, and returns a new product code.

What is the correct DataWeave code to define the newProdCode function?

- A. `fun newProdCode(itemID: Number, productCategory: String) --> "PC-" ++ productCategory ++ (itemID as String)`
- B. `fun newProdCode(itemID: Number, productCategory: String) = "PC-" ++ productCategory ++ (itemID as String)`
- C. `function newProdCode(itemID: Number, productCategory: String) = "PC-" ++ productCategory ++ (itemID as String)`
- D. `var newProdCode(itemID: Number, productCategory: String) -> "PC-" ++ productCategory ++ (itemID as String)`

Correct Answer: B

[Latest MCD-LEVEL1 Dumps](#)

[MCD-LEVEL1 PDF Dumps](#)

[MCD-LEVEL1 Exam Questions](#)