

# 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.leads4pass.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**

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

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

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

A. 

```
/department:  
  get:  
    uriParameter:  
      storeId:
```

B. 

```
get:  
  queryParameters:  
    department:  
  uriParameters:  
    storeId:
```

C. 

```
/{storeId}:  
  get:  
    queryParameters:  
      department:
```

D. 

```
get:  
  uriParameters:  
    {storeId}:  
  queryParameters:  
    department:
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

---

**QUESTION 2**

What is output of Dataweave flatten function?

A. Object

B. Map

C. Array

D. LinkedHashMap

Correct Answer: C

Correct answer is Array.

Flatten turns a set of subarrays (such as [ [1,2,3], [4,5,6], [], [null] ]) into a single, flattened array (such as [ 1, 2, 3, 4, 5, 6, null ]). This example defines three arrays of numbers, creates another array containing those three arrays, and then

uses the flatten function to convert the array of arrays into a single array with all values.

Source

```
%dw 2.0
```

```
output application/json
```

```
var array1 = [1,2,3]
```

```
var array2 = [4,5,6]
```

```
var array3 = [7,8,9]
```

```
var arrayOfArrays = [array1, array2, array3]
```

```
flatten(arrayOfArrays) Output [ 1,2,3,4,5,6,7,8,9 ]
```

---

### QUESTION 3

Refer to the exhibits. A company has defined this Book data type and Book example to be used in APIs. What is valid RAML for an API that uses this Book data type and Book example?

```

#%RAML 1.0 DataType          # %RAML 1.0 NamedExample
# bookDataType.raml         # bookExample.raml

type: object                bookExample:
properties:                 |
  ID?: integer              | ID: 101
  title: string             | title: Shakespeare
  author: string            | author: Encyclopaedia Britannica
  publisher?: string        | publisher: John Wiley & Sons
  year: integer             | year: 2007
  ISBN:                     | ISBN: "0471767840"
  type: string
  required: true

```

A.

```

#%RAML 1.0
title: Books

Book: BookDataType.raml

/books:
  post:
    body:
      application/json:
        type: Book
        examples:
          input: BookExample.raml
    responses:
      201:
        body:
          application/json:
            example:
              message: Book added

```

B.

```

#%RAML 1.0
title: Books

Book: !include BookDataType.raml

/books:
  post:
    body:
      application/json:
        type: Book
        examples:
          input: !include BookExample.raml
    responses:
      201:
        body:
          application/json:
            example:
              message: Book added

```

C.

```

#%RAML 1.0
title: Books

types:
  Book: ABC/DataTypes/BookDataType.raml

/books:
  post:
    body:
      application/json:
        type: Book
        examples:
          input: ABC/Examples/BookExample.raml
    responses:
      201:
        body:
          application/json:
            example:
              message: Book added

```

D.

```

#%RAML 1.0
title: Books

types:
  Book: !include BookDataType.raml

/books:
  post:
    body:
      application/json:
        type: Book
        examples:
          input: !include BookExample.raml
    responses:
      201:
        body:
          application/json:
            example:
              message: Book added

```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

---

#### QUESTION 4

What MuleSoft API-led connectivity layer is intended to expose part of a backend database without business logic?

A. Data layer

B. Process layer

C. Experience layer

D. System layer

Correct Answer: D

Correct answer is System layer

System APIs provide a means for insulating the data consumers from the complexity or changes to the underlying backend systems.

MuleSoft recommends three-layered approach to API-led connectivity, highlighting the three layers:

1.

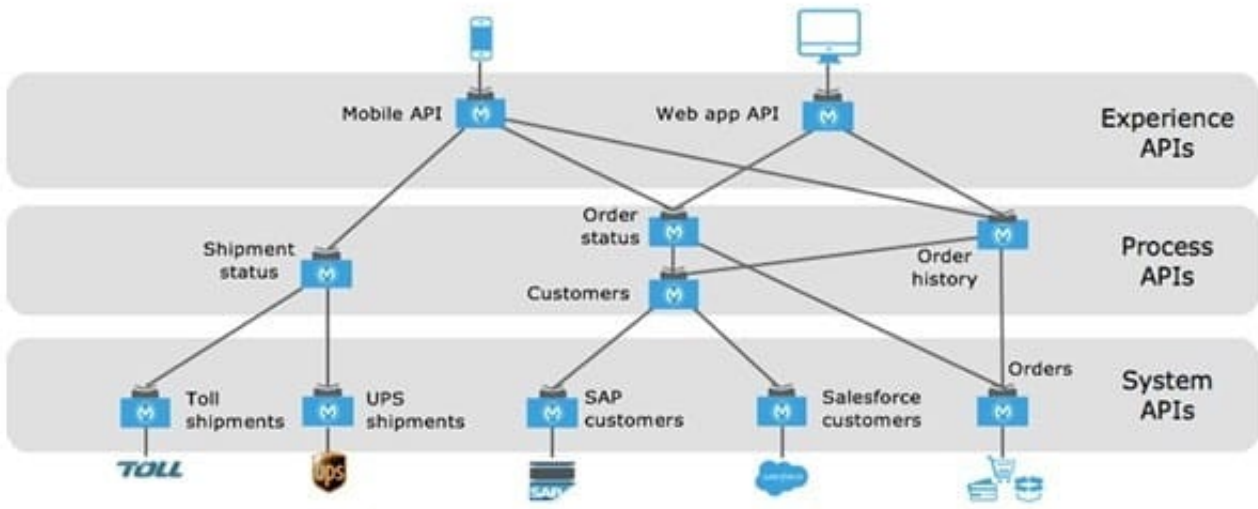
System APIs

2.

Process APIs

3.

Experience APIs System APIs are the core systems of record underlying core systems of record (e.g. ERPs, key customer and billing systems, databases, etc.). Process APIs allow you to define a common process which the organization can share, and these APIs perform specific functions, provide access to non-central data, and may be built by either Central IT or Line of Business IT. And finally, the Experience APIs are the means by which data can be reconfigured so that it is most easily consumed by its intended audience, all from a common data source. The three-layered structure allows for a seamless flow of data from systems of record to new experiences, and allows for reusability of assets rather than point to point connections. This approach provides a distributed and tailored approach to architecture, greater flexibility through loose coupling, and deeper operational visibility into what is being built.



## QUESTION 5

Refer to the exhibit.



The default scope in choice router recursively calls the color flow.

A web client sends a PUT request to the HTTP listener with payload Blue. What response is returned to the web client?

- A. "Blk"
- B. "Green"
- C. A timeout error
- D. ["Blue", "Red", "Blk"]

Correct Answer: C

Sequence can be described as follows.

1.

When web client sends the request , it sends payload as Blue (mentioned in question)

2.

In first iteration this will go to default choice , which sets payload to Red

3.

Via default route , flow will call itself with payload as Red

4.

In second iteration , as payload is Red , it will go to first route in choice router which will set payload to Blk and second iteration will end returning back payload as Blk to first iteration.

5.

This Blk will be again set to payload and first iteration will end returning back response as Blk

[Latest MCD-LEVEL1 Dumps](#)

[MCD-LEVEL1 Practice Test](#)

[MCD-LEVEL1 Braindumps](#)