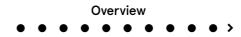Take this multiple-choice quiz to assess your mastery of core concepts taught in the *Anypoint Platform Development: Fundamentals (Mule 4)* (/instructor-led-training/apdev-fundamentals4), the self-study *MuleSoft.U Development Fundamentals (Mule 4)* (https://training.mulesoft.com/course/mulesoftu-fundamentals4), the *Anypoint Platform Development: Mule 4 for Mule 3 Users* (/instructor-led-training/apdev-mule443), and the self-study *MuleSoft.U Mule 4 for Mule 3 Users* (https://training.mulesoft.com/course/mulesoftu-mule443) courses and to help you prepare for the *MuleSoft Certified Developer - Level 1 (Mule 4)* (https://training.mulesoft.com/exam/mcd-level1) or *MuleSoft Certified Developer - Level 1 (Mule 4) DELTA* (https://training.mulesoft.com/exam/mcd-level1-delta) certification exam.

- 5+ multiple-choice questions for each topic
- Identifies strengths and weaknesses
- Comparable difficulty to the proctored exam

Overview

● ● ● ● ● ● ● ● ● ● ❯

Designing APIs

Quiz

You got 4 / 5. Please review your answers.

Close.

1.

An API has been created in Design Center. What is the next step to make the API discoverable?

○ Deploy the API to a Maven repository

○ Enable autodiscovery in API Manager

○ Publish the API from inside flow designer

◉ Publish the API to Anypoint Exchange

> *Correct!*

2.

Refer to the exhibit. What is the correct URL to submit a GET request to /patients?

```
1    #%RAML 1.0
2    title: ACME Medical API
3    baseUri: http://dev.acme.com/api
4
5    /patients:
6      get:
7        queryParameters:
8          year:
9            type: integer
10           example: 2017
```

○ http://dev.acme.com/patients?year=2016

○ http://dev.acme.com/api/patients

○ http://dev.acme.com/patients

◉ http://dev.acme.com/api/patients?year=2016

> *Correct!*

3.

Wrapping Up the Course

←

A RAML example fragment named BankAccountsExample.raml is placed in the examples folder in an API specification project.

What is the correct syntax to reference the fragment?

- ○ examples: !include BankAccountsExample.raml
- ○ examples: #import BankAccountsExample.raml
- ● examples: !include examples/BankAccountsExample.raml
- ○ examples: #import examples/BankAccountsExample.raml

> *Correct!*

4.

Refer to the exhibit. There is an error in the flight_id resource's GET method. What needs to be done to fix the problem?

```
1. #%RAML 1.0
2. version: v1
3. title: American Flights API
4.
5. /flights:
6.   get:
7.
8.   /{flight_id}:
9.
10. get:
```

- ○ Remove blank line on row 9
- ○ Enclose flight_id in parentheses () instead of curly braces {}
- ● Indent the get method under the {flight_id} resource
- ○ Outdent the {flight_id} resource
- ○ Remove the curly braces {} around flight_id

> *Correct!*

5.

Refer to the exhibit. This RAML specification includes an XML example that matches the Records data type defined in another RAML file named recordsDataType.raml.

Using the Records type, how can this XML example be represented in RAML?

```raml
#%RAML 1.0
title: Records API

/records:
  get:
    responses:
      200:
        body:
          application/xml:
            example: |
              <music>
                <collection>Deep Collection</collection>
                <artists>
                  <artist1>Deep Purple</artist1>
                  <artist2>Rainbow</artist2>
                </artists>
              </music>
```

Answers

A.

```raml
#%RAML 1.0
title: Records API

types:
  Records: !include recordsDataType.raml

/records:
  get:
    responses:
      200:
        body:
          application/xml:
            type: Records
            example:
              music:
                collection: "Deep Collection "
                artists:
                  - artist1: "Deep Purple"
                  - artist2: "Rainbow"
```

B.

```raml
#%RAML 1.0
title: Records API

types:
  Records: !include recordsDataType.raml

/records:
  get:
    responses:
      200:
        body:
          application/xml:
            type: Records
            example:
              music:
                collection: "Deep Collection "
                artists:
                  "Deep Purple"
                  "Rainbow"
```

C.

```
#%RAML 1.0
title: Records API

types:
  Records: !include recordsDataType.raml

/records:
  get:
    responses:
      200:
        body:
          application/xml:
            type: Records
            example:
              music:
                collection: "Deep Collection "
                artists: "Deep Purple", "Rainbow"
```

D.

```
#%RAML 1.0
title: Records API

types:
  Records: !include recordsDataType.raml

/records:
  get:
    responses:
      200:
        body:
          application/xml:
            type: Records
            example:
              music:
                collection: "Deep Collection "
                artists:
                  artist1: "Deep Purple"
                  artist2: "Rainbow"
```

- ○ A
- ○ B
- ○ C
- ○ D

Sorry, that's incorrect.

You got 4 / 5. Please scroll up to review your answers. Close.

Next: Building APIs