



## Welcome to Cert007 - Your Ultimate IT Certification Partner



➤ Real Exam Questions

➤ Free Updates

➤ Expert Support

➤ Instant Access

➤ Money-Back Guarantee



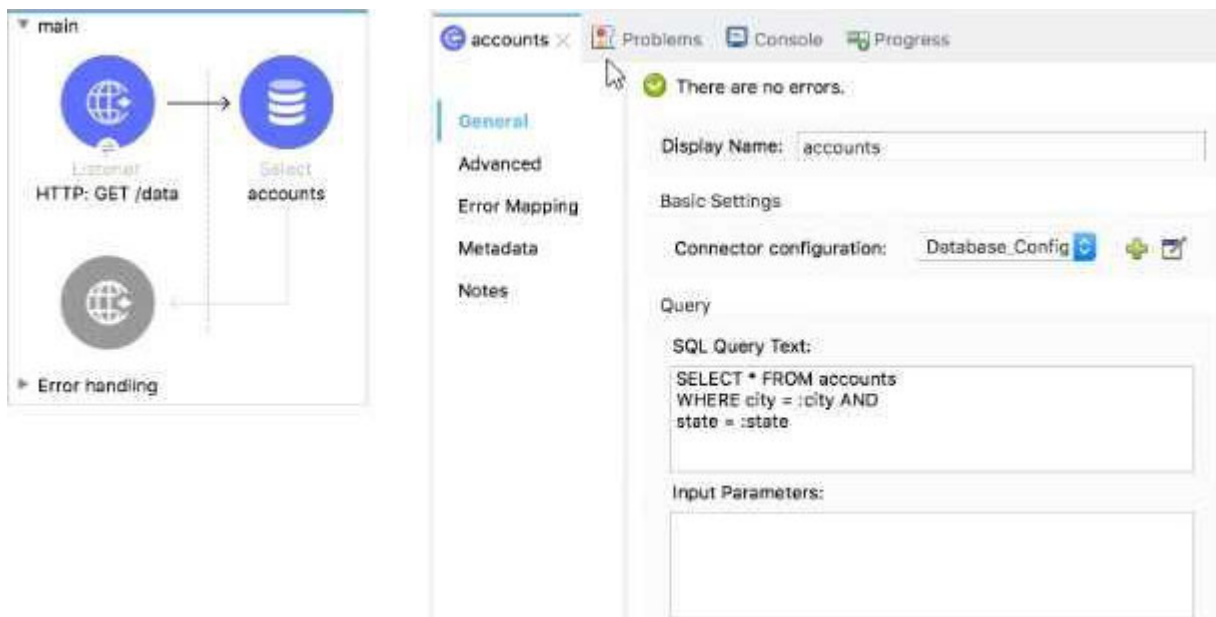
Visit us at <https://www.cert007.com/> for more information

**Exam :** **MuleSoft Developer I**

**Title :** **Salesforce Certified  
MuleSoft Developer I**

**Version :** **DEMO**

1.Refer to the exhibit.



What expression correctly specifies input parameters to pass the city and state values to the SQL query?

A)

```
#[
  {
    city: "San Francisco",
    state: "CA"
  }
]
```

B)

```
#[
  [
    "San Francisco",
    "CA"
  ]
]
```

C)

```
#[
  inputParams: {
    city: "San Francisco",
    state: "CA"
  }
]
```

D)

```
#[
  inputParams: [
    "San Francisco",
    "CA"
  ]
]
```

A. Option A

B. Option B

C. Option C

D. Option D

**Answer: A**

**Explanation:**

MuleSoft Documentation

Reference: <https://docs.mulesoft.com/db-connector/1.9/database-connector-select>

2.A Mule flow has three Set Variable transformers.

What global data structure can be used to access the variables?

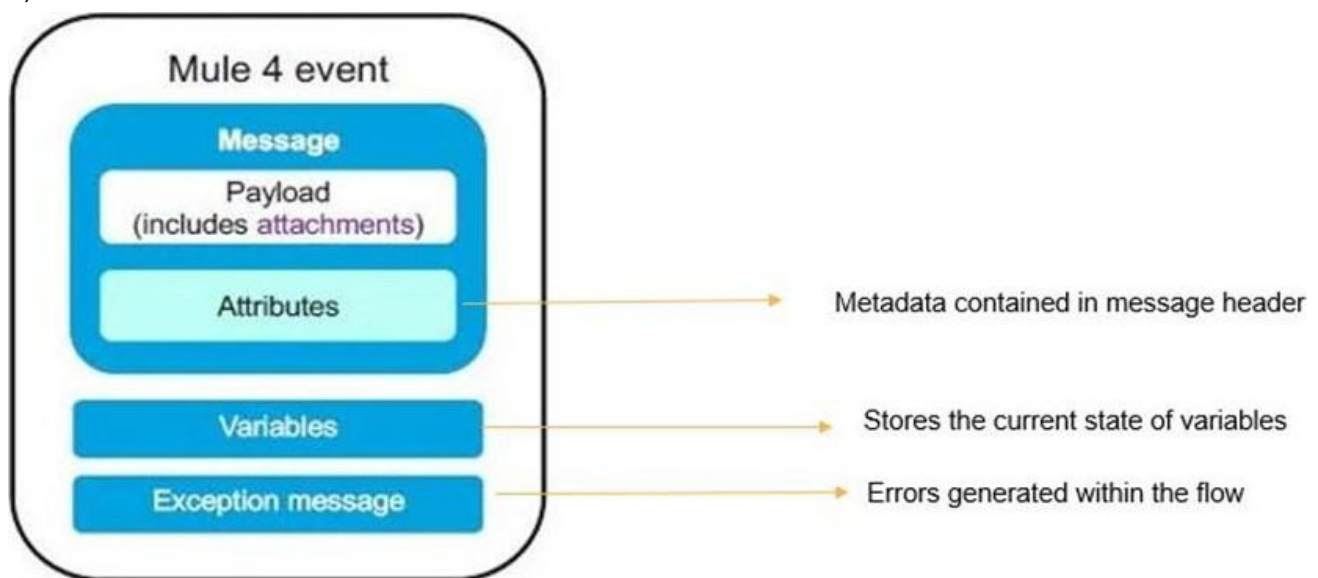
- A. Mule event attributes
- B. Mule event message
- C. Mule application properties
- D. Mule event

**Answer: D**

**Explanation:**

Mule event is correct answer. Mule event has two parts which are as follows

- 1) Message (which contains payload and attributes like headers and query/uri parameters)
- 2) Variables



3.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

**Answer: D**

**Explanation:**

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

4. Refer to the exhibit.



```
<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 to a web client request to `http://localhost:8081`?

- A. After  
B. before  
C. Validation Error  
D. null

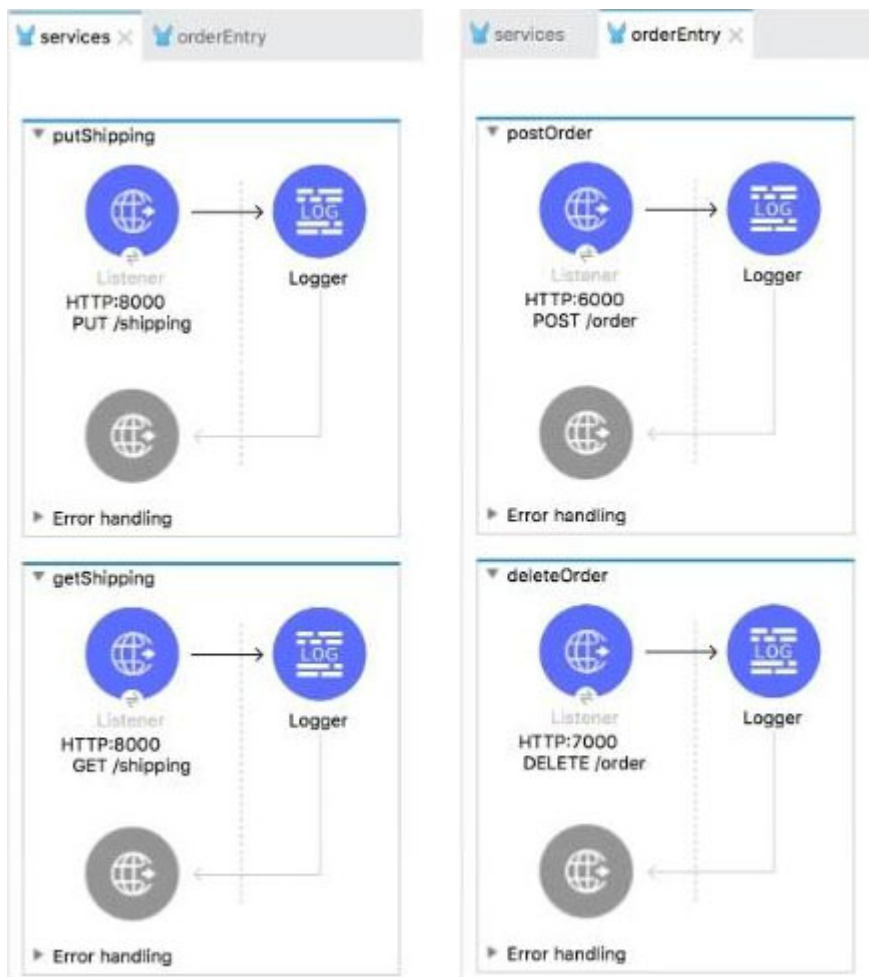
**Answer: C**

**Explanation:**

The screenshot displays the MuleSoft IDE interface with three main components:

- Message Flow Diagram:** A flowchart for 'question12Flow' showing a sequence of steps: 'Listen HTTP-GET' (purple circle), 'Set Payload "Before"' (green circle), 'to url payload' (blue circle), 'Set Payload "After"' (green circle), and 'Logger' (purple circle). An 'Error handling' section is also visible below the main flow.
- Advanced REST client:** A window showing a GET request to 'http://localhost:8086/q12'. The response status is '500 Server Error' with a time of '9941.33 ms'. Below the response, there are buttons for 'COPY', 'SAVE', and 'SOURCE VIEW'. A red circle highlights the text 'Validation Error' in the response body.
- Console:** A window showing the log output for 'question12 [Mule Applications] Mule Server 4.2.1 EE'. The log includes an error message: 'Validation Error. Error type: VALIDATION:NOT\_NULL. Element: question12Flow/processorgw/ 8 question12:question12.xml:14 (payload). Element XML: <validation:is-null docname="payload" docid="3d0272d6-8a27-46ea-bf46-b0c26be600b" value="#[payload]" message="Validation Error"></validation:is-null>'. The log also includes a debug level logging instruction: '(set debug level logging or '!-Debug.verbose.exceptions=true' for everything)'.

5. Refer to the exhibits.



The two Mule configuration files belong to the same Mule project. Each HTTP Listener is configured with the same host string and the port number, path, and operation values are shown in the display names. What is the minimum number of global elements that must be defined to support all these HTTP Listeners?

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

**Answer: B**

**Explanation:**

In this case three configurations will be required each for port 8000, 6000 and 7000.

There would be three global elements defined for HTTP connections.

Each HTTP connection will have host and port. One example shown below with host as localhost and port 6000

The screenshot shows a 'Global Element Properties' dialog box for an 'HTTP Listener config'. The dialog has a title bar with a close button. Below the title, it says 'Configuration element for a HttpListener.' There are three tabs: 'General', 'Notes', and 'Help'. The 'General' tab is selected. Under the 'Name' field, the value 'HTTP\_Listener\_config' is entered. Below this is a 'Connection' section with its own sub-dialog box. This sub-dialog has three tabs: 'General', 'TLS', and 'Advanced'. The 'General' tab is selected. It contains three fields: 'Protocol' set to 'HTTP (Default)', 'Host' set to 'All Interfaces [0.0.0.0] (default)', and 'Port' set to '6000'. At the bottom of the main dialog are three buttons: a help button (question mark icon), 'Test Connection...', and 'OK' (highlighted with a blue border), and a 'Cancel' button.

Global Element Properties

### HTTP Listener config

Configuration element for a HttpListener.

General Notes Help

Name: HTTP\_Listener\_config

Connection

General TLS Advanced

Connection

Protocol: HTTP (Default)

Host: All Interfaces [0.0.0.0] (default)

Port: 6000

Test Connection... OK Cancel

To use an HTTP listener, you need to declare a configuration with a corresponding connection. This declaration establishes the HTTP server that will listen to requests.

Additionally, you can configure a base path that applies to all listeners using the configuration.

```
<http:listener-config name="HTTP_Listener_config" basePath="/api/v1"> <http:listener-connection  
host="0.0.0.0" port="8081" />
```

```
</http:listener-config>
```

<https://docs.mulesoft.com/http-connector/1.6/http-listener-ref#http-listener-configuration>