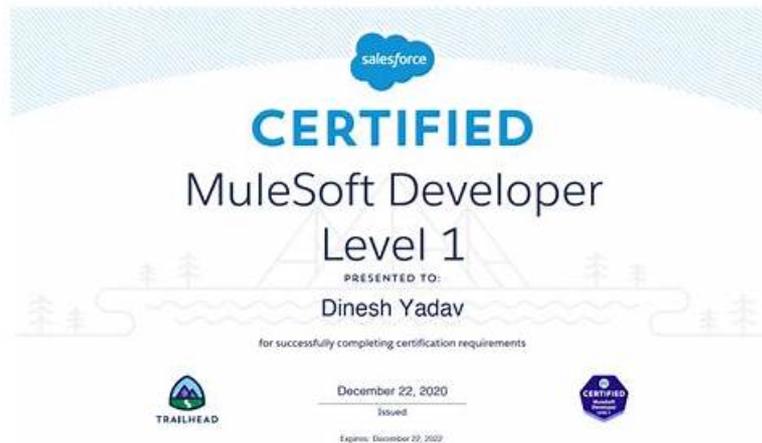


# Valid Salesforce-MuleSoft-Developer-I Test Book | Salesforce-MuleSoft-Developer-I PDF Guide



P.S. Free 2025 Salesforce Salesforce-MuleSoft-Developer-I dumps are available on Google Drive shared by DumpsKing: <https://drive.google.com/open?id=19FGLAc5SepvBd1BJVlycbGNhAGtqAvnz>

Experts at DumpsKing have also prepared Salesforce Salesforce-MuleSoft-Developer-I practice exam software for your self-assessment. This is especially handy for preparation and revision. You will be provided with an examination environment and you will be presented with actual exam Salesforce Salesforce-MuleSoft-Developer-I Exam Questions. This sort of preparation method enhances your knowledge which is crucial to excelling in the actual Salesforce Salesforce-MuleSoft-Developer-I certification exam.

We offer you free update for one year for Salesforce-MuleSoft-Developer-I study guide, namely, in the following year, you can obtain the latest version for free. And the latest version for Salesforce-MuleSoft-Developer-I exam dumps will be sent to your email automatically. In addition, Salesforce-MuleSoft-Developer-I exam materials are high quality, since we have experienced experts to compile and verify them, therefore the quality and accuracy can be guaranteed, so you can use them at ease. We have online and offline chat service, and if you have any questions about Salesforce-MuleSoft-Developer-I Exam Dumps, you can consult us, and we will give you reply as quickly as possible.

>> Valid Salesforce-MuleSoft-Developer-I Test Book <<

## Key Features of Salesforce Salesforce-MuleSoft-Developer-I PDF Questions By DumpsKing

As far as our Salesforce-MuleSoft-Developer-I practice test is concerned, the PDF version brings you much convenience with regard to the following two aspects. On the one hand, the PDF version contains demo where a part of questions selected from the entire version of our Salesforce-MuleSoft-Developer-I test torrent is contained. In this way, you have a general understanding of our actual prep exam, which must be beneficial for your choice of your suitable exam files. On the other hand, our Salesforce-MuleSoft-Developer-I Preparation materials can be printed so that you can study for the exams with papers and PDF version. With such benefits, why don't you have a try?

## Salesforce Salesforce-MuleSoft-Developer-I Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none"><li>Processing Records: Processing records includes methods for processing individual records in a collection and explaining how Mule events are processed by the For Each scope. It also involves using the Batch Job with Batch Steps and a Batch Aggregator.</li></ul>

Topic 2	<ul style="list-style-type: none"> <li>Handling Errors: Handling errors includes describing default error handling in Mule applications and defining custom global default error handlers. It involves comparing On Error Continue and On Error Propagate scopes, creating error handlers for a flow, using the Try scope, and mapping errors to custom application errors.</li> </ul>
Topic 3	<ul style="list-style-type: none"> <li>Debugging and Troubleshooting Mule Applications: Using breakpoints to inspect a Mule event during runtime, installing missing Maven dependencies, and reading and deciphering Mule log error messages are sub-topics of this topic.</li> </ul>
Topic 4	<ul style="list-style-type: none"> <li>Creating Application Networks: The topic of creating Application Networks encompasses understanding MuleSoft's proposal for closing the IT delivery gap and describing the role and characteristics of the modern API. It also includes the purpose and roles of a Center for Enablement (C4E), and the benefits of API-led.</li> </ul>
Topic 5	<ul style="list-style-type: none"> <li>Building API Implementation Interfaces: This topic involves manually creating a RESTful interface for a Mule application and generating a REST Connector from a RAML specification. It also includes describing the features and benefits of APIKit.</li> </ul>
Topic 6	<ul style="list-style-type: none"> <li>Transforming Data with DataWeave: It involves writing DataWeave scripts and using DataWeave functions. This topic also includes defining and using DataWeave variables, functions, and modules, and applying correct syntax.</li> </ul>
Topic 7	<ul style="list-style-type: none"> <li>Accessing and Modifying Mule Events: It describes the Mule event data structure. Moreover, the topic focuses on usage of transformers and enriching Mule events.</li> </ul>
Topic 8	<ul style="list-style-type: none"> <li>Routing Events: It focuses on using the Choice router for conditional logic and the Scatter-Gather router to multicast events. This topic also involves validating data by using the Validation module.</li> </ul>
Topic 9	<ul style="list-style-type: none"> <li>Using Connectors: It focuses on retrieving data from REST services using HTTP Request or REST Connector. Moreover, the topic covers using a Web Service Consumer connector for SOAP web services and the Transform Message component.</li> </ul>
Topic 10	<ul style="list-style-type: none"> <li>Designing APIs: Designing APIs involves describing the lifecycle of the modern API and using RAML to define various aspects of an API. It includes identifying when to use query parameters vs URI parameters, and defining API parameters.</li> </ul>

## Salesforce Certified MuleSoft Developer (Mule-Dev-201) Sample Questions (Q18-Q23):

### NEW QUESTION # 18

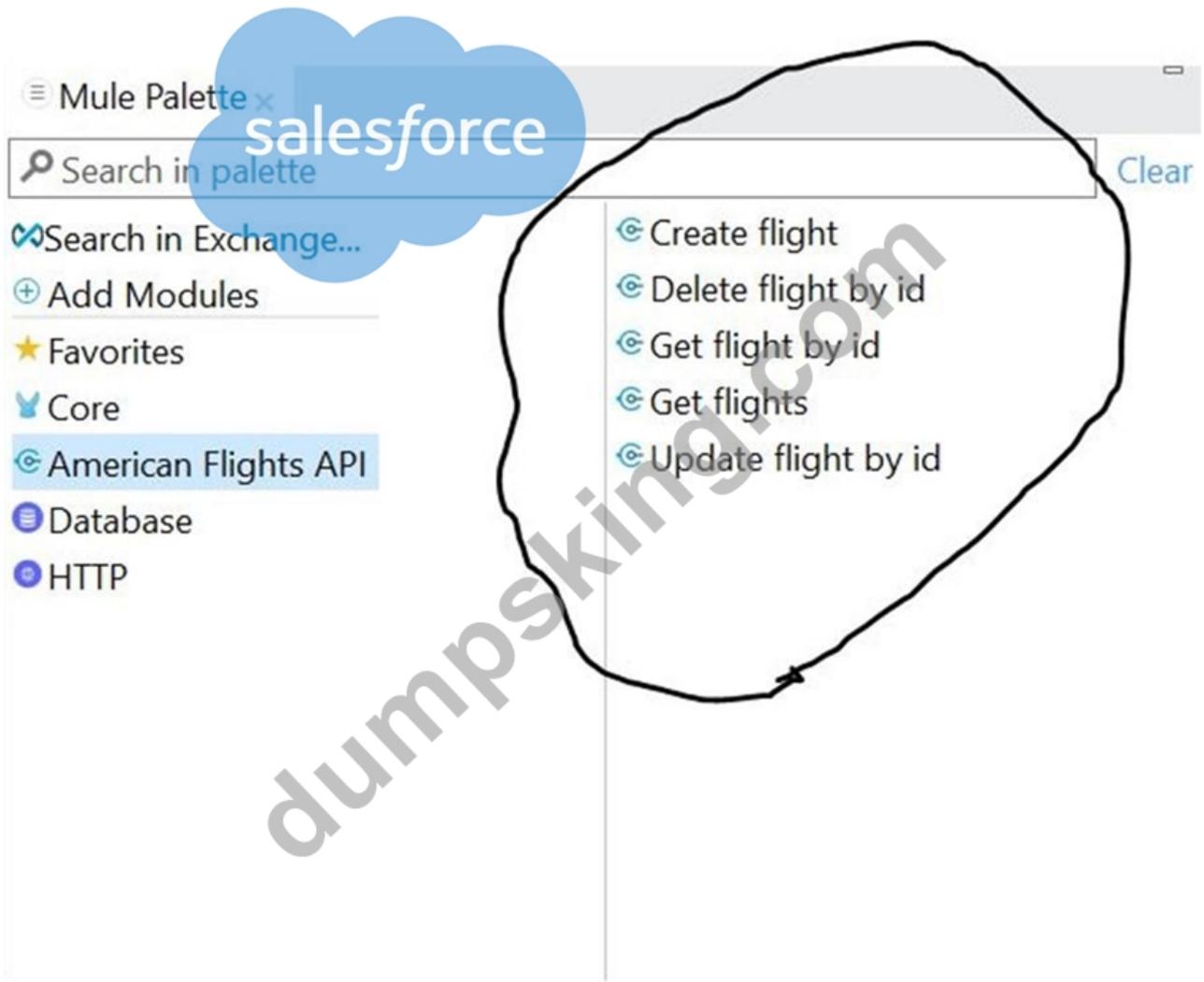
A REST connect module is generated for a RAML specification, and then the rest connect module is imported in mule application in Anypoint Studio. For each method of the RAML specification, what does the REST connect module provide?

- A. An event source
- B. A scope
- C. A flow
- D. An operation

**Answer: C**

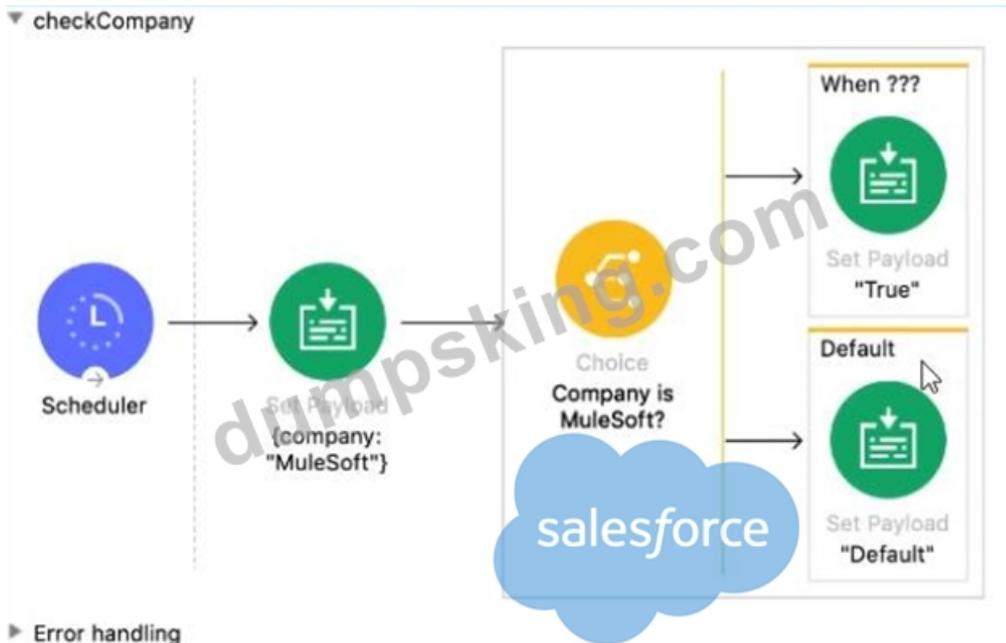
Explanation:

Correct answer is an operation. For each method of the RAML specification, REST connect module provide an operation. Please refer to the below screenshot.



**NEW QUESTION # 19**

Refer to the exhibits.



```

<flow name="checkCompany">
  <scheduler doc:name="Scheduler">
    <scheduling-strategy>
      <fixed-frequency frequency="5000" />
    </scheduling-strategy>
  </scheduler>
  <set-payload value='#[{company: "MuleSoft"}]' doc:name="{company: "MuleSoft"}' />
  <choice doc:name="Company is MuleSoft?">
    <when expression="When ???">
      <set-payload value='#[ "True" ]' doc:name="True" />
    </when>
    <otherwise>
      <set-payload value='#[ "Default" ]' doc:name="Default" />
    </otherwise>
  </choice>
</flow>

```

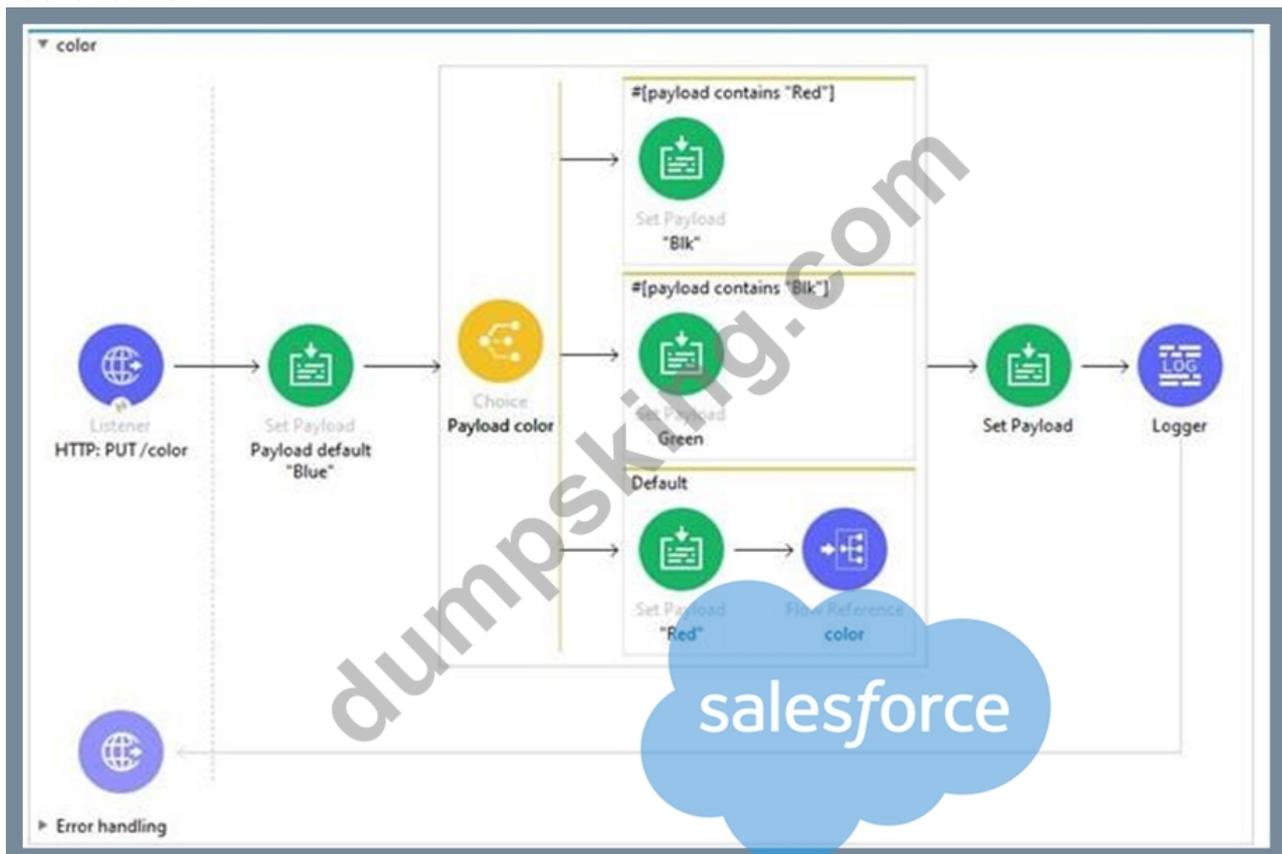
The <when> expression for the Choice router needs to be written.  
 What is a valid <when> expression to route Mule events to the non-default flow?

- A. #[ if( company = "MuleSoft") ]
- B. #[ 'MuleSoft' == payload.company ]
- C. #[ company = "MuleSoft" ]
- D. #[ if( 'MuleSoft' == payload.company ) ]

Answer: B

#### NEW QUESTION # 20

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

**Answer: C**

Explanation:

Sequence can be described as follows.

- \* When web client sends the request , it sends payload as Blue (mentioned in question)
- \* In first iteration this will go to default choice , which sets payload to Red
- \* Via default route , flow will call itself with payload as Red
- \* 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.
- \* This Blk will be again set to payload and first iteration will end returning back response as Blk

#### **NEW QUESTION # 21**

A RAML example fragment named StudentExample.raml is placed in the examples folder in an API specification project. What is the correct syntax to reference the fragment?

- A. examples: !include StudentExample.raml
- B. examples: #import examples/StudentExample.raml
- C. examples: !include examples/StudentExample.raml
- D. examples: #import StudentExample.raml

**Answer: C**

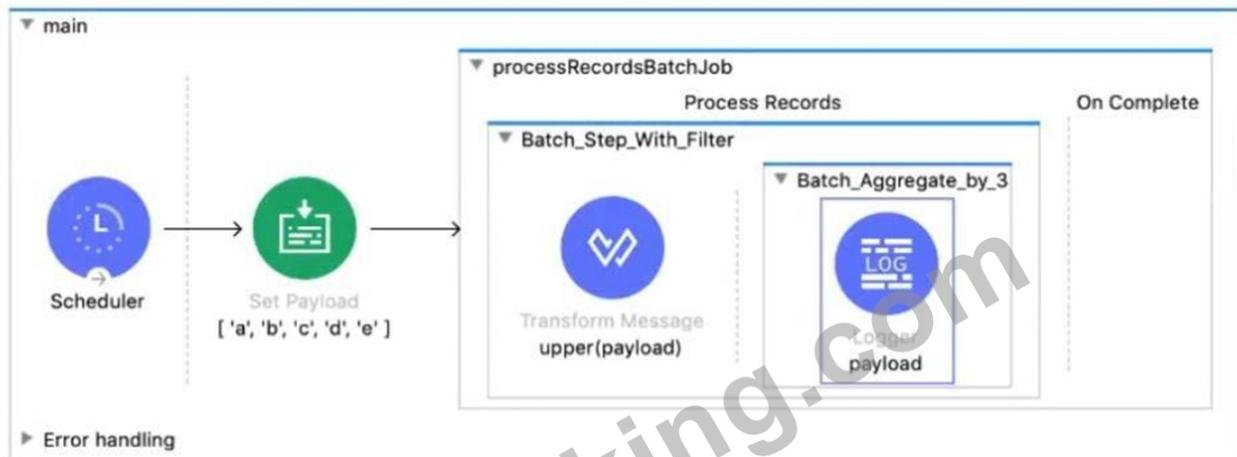
Explanation:

To include property. To keep the API definition concise, you can include external content, such as documentation, schemas, and frequently used patterns outside the definition itself. The parser interprets !include as if the content of the externally-hosted file or a URL were declared in-line.

To use the fragments in RAML you have to include the exact path(copy the path) of that fragment you want to use as shown below  
Option 3 is the correct as correct syntax is examples: !include examples/StudentExample.raml

#### **NEW QUESTION # 22**

Refer to the exhibits.



```

<flow name="main" >
  <scheduler doc:name="Scheduler" > <scheduling-strategy >
    <fixed-frequency frequency="10000"/></scheduling-strategy> </scheduler>
    <set-payload value="#[[ 'a', 'b', 'c', 'd', 'e' ]]" doc:name="[ 'a', 'b', 'c', 'd', 'e' ]" />
    <batch:job jobName="processRecordsBatchJob" >
      <batch:process-records >
        <batch:step name="Batch_Step_With_Filter"
          acceptExpression="#[not (payload contains "b") ]">
          <ee:transform doc:name="upper(payload)"><ee:message >
            <ee:set-payload ><! [CDATA[%dw 2.0
              output application/json
              ---
              upper(payload)]]></ee:set-payload>
            </ee:message></ee:transform>
          <batch:aggregator doc:name="Batch_Aggregate_by_3" size="3">
            <logger level="INFO" doc:name="payload"
              message="#[output application/json --- payload]" />
          </batch:aggregator>
        </batch:step>
      </batch:process-records>
    </batch:job>
  </flow>

```



The input array of strings is processed by the batch job that processes, filters, and aggregates the values. What is the last message logged by the Logger component after the batch job completes processing?

- A. [ "A", "C", "D" ], [ "E" ]
- **B. [ "E" ]**
- C. [ "A", "C", "D", "E" ]
- D. [ "D", "E" ]

**Answer: B**

Explanation:

Logs would look like:

```
INFO 2021-06-09 19:14:56,039[[MuleRuntime].uber.06: [validationtest].batch-job-validationtestBatch_Job- work-manager @6de10f3e] [processor:validationtestFlow/processors/1/route/0/route/0/aggregator/processors /0; event: bfb751e1-9939-11eb-9f69-02053763653a] org.mule.runtime.core.internal.processor.
```

LoggerMessageProcessor:

```
[
  "\A",
  "\C",
  "\D"
]
```

```
INFO 2021-06-09 19:15:02,486 [[MuleRuntime].uber.06: [validationtest].batch-job-validationtestBatch_Job- work-manager @6de10f3e] [processor: validationtestFlow/processors/1/route/0/route/0/aggregator/processors /0; event: bfb751e1-9939-11eb-9f69-02053763653a] org.mule.runtime.core.internal.processor.
```

LoggerMessageProcessor:

```
["E"]
```

Batch aggregator value is 3. Hence in first time it will print [ "A", "C", "D" ] and in next iteration it will print [ "E" ]

----- Correct answer is [ "E" ]



BONUS!!! Download part of DumpsKing Salesforce-MuleSoft-Developer-I dumps for free: <https://drive.google.com/open?id=19FGLAc5SepvBd1BJVlycbGNhAGtqAvnz>