

MuleSoft.MCD-Level-1.v2022-10-15.q173

□□□□:	MCD-Level-1
□□□□:	MuleSoft Certified Developer - Level 1 (Mule 4)
□□□:	MuleSoft
□□ □□ □□□:	173
□□:	v2022-10-15
# □□ □:	1801
# □□ □□□:	1730
https://www.krdump.com/MuleSoft.MCD-Level-1.v2022-10-15.q173.html	

NEW QUESTION: 1

□□□□ □□□□□.

□ □□□



□ □ □ □

The screenshot shows the MuleSoft Anypoint Studio interface. On the left, the 'order.xml' file is open, displaying the following XML content:

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

On the right, the 'Output Payload' window shows the resulting JSON:

```
[
  {
    "index": 0,
    "orderId": "592",
    "itemName": "T-shirt Navy",
    "lineItemPrice": 20
  },
  {
    "index": 1,
    "orderId": "972",
    "itemName": "Cargo Shorts",
    "lineItemPrice": 60
  }
]
```

□□□ □□□□ □□□□ □□□□□□ □□□□ □□□□□□?

A)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.orderId,
  itemName: value.item,
  lineItemPrice: (value.price as :number) * (value.quantity as :number)
})
```

B)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.@orderId,
  itemName: value.item,
  lineItemPrice: (value.price as Number) * (value.quantity as Number)
})
```

C)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.@orderId,
  itemName: value.item,
  lineItemPrice: (value.price as :number) * (value.quantity as :number)
})
```

D)

```
payload.order.*item map( (value,index) -> {
  index: index,
  orderId: value.orderId,
  itemName: value.item,
  lineItemPrice: (value.price as Number) * (value.quantity as Number)
})
```

- A. 10 D
- B. 10 A
- C. 10 B
- D. 10 C

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 3

API endpoint returns a list of patients for a given year?
 http://dev.acme.com/api/patients?year=2021

- A. 2021 is a required parameter
- B. 2021 is an optional parameter
- C. year is a required parameter
- D. year is an optional parameter

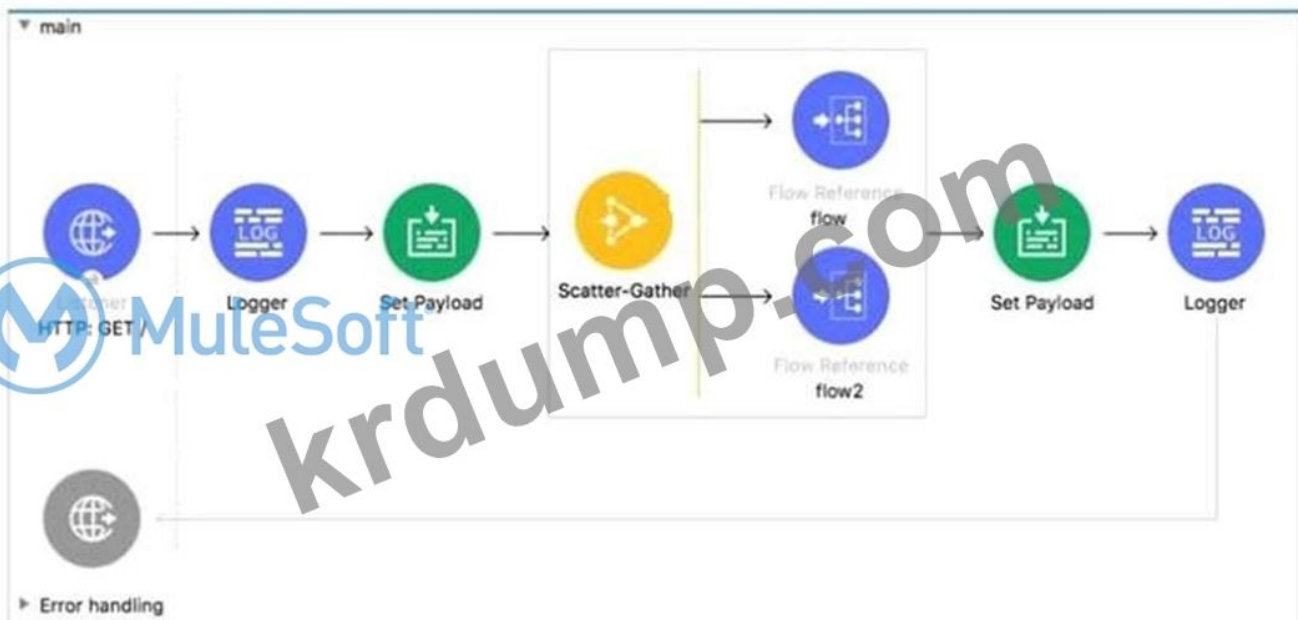
Answer: B ([LEAVE A REPLY](#))

2021 is an optional parameter.

year is a required parameter in the uri. The parameter is optional in the API endpoint. The parameter is optional in the API endpoint.

NEW QUESTION: 4

Scatter-Gather connector is used to...



Scatter-Gather connector is used to parallelize the flow. The Scatter-Gather connector is used to parallelize the flow. The Scatter-Gather connector is used to parallelize the flow.

- A. 10
- B. 50
- C. 40
- D. 20

Answer: C ([LEAVE A REPLY](#))

Scatter-Gather □ □□□ □ □□□ □□□ □□□□. □□□ □ □□ □ □□□ □□□ □□□□ □. □□□ □□□ □□□□ □ □□□ □ □□□ □□ 3□ 40□□□□.

NEW QUESTION: 5

□ □□ □□□ □□ □□ □□ □□□□. □□ □□□□□□□□ □□□□ □□□ □□□□□ □ □□ API□ □□□ □ 2□□□ □□□□□. □□□ □ □□ □□ □□ □□□ □□ □□□ □□ □□□ API□ □□ □□□□□ □□□ □□ □□□□□.

MuleSoft □ □□□ □□ □□ □□□ □ □□ □□ □□□ 2□□ □□□ □ □□□□□?

- A. □□ □□
- B. □□□ □□
- C. MuleSoft □□ □□
- D. □□ API □□□□□

Answer: (SHOW ANSWER)

Center for Enablement □ □□□□□. □□□□□ □□ IT, LOB(□□ □□) □□, □□□ □□ □□ □□□□□ □□□ □□□ □□□ □□□ □□□ □□□ □□ □□□ □□□, □□ □ □□□□ □ □□ □□□□□. □ □□ □ □□ □□ API□ □□□□ □□ Center for Enablement□ □□ □□ □□ □□□□. □□□ □□□□ □□ □ □□□□□.

NEW QUESTION: 6

□□ □ API Notebooks □ □□□ □□ □□ □□□□□?

- A. API □□
- B. API□ □□□□□ □□
- C. API□ □□ □□□ □□
- D. □□ □□□□ □□□ □□□ □□ □□

Answer: (SHOW ANSWER)

□□□ API□ □□ □□□ □□□□□.

API Notebook □ API □□, □□□ API □□□ □ □□ □□□ □□ □□ □□□ □□ □□ □ □□ □□□□, API □□□□□□ □□□□□□□□□□. API Notebook □ □□□□ □□□ □□□ □□□ □□ □ □□ □□□□ □□□ □□□ □ □□□□. □□□ API□ □□ □□ □□□□□ □□□ □ □□□□.

MuleSoft □□ □□ : <https://docs.mulesoft.com/api-manager/1.x/api-notebook-concept>

NEW QUESTION: 7

□□□ □□□□□.



```

<flow name="main">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <http:request method="POST" doc:name="HTTP: POST /data" url="http://localhost:8081/data"/>
  <jms:publish-consume doc:name="JMS: num1" config-ref="JMS_Config" destination="num1"/>
  <jms:publish doc:name="JMS: num2" config-ref="JMS_Config" destination="num2"/>
  <set-payload value="#[payload + 1]" doc:name="payload + 1" />
</flow>
  
```

http://localhost.8081/□ □□ □□□□ □□□□ □□□□
 □□□□
 http://localhost;8081/□ □□ □□□□ □□ □□□□□ □□□□□?

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

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 8

DataWeave□□ □ □□□ □□□ □ □□ □□□□ □□□□□?

- A. □□
- B. □□□□
- C. □□
- D. □□

Answer: (SHOW ANSWER)

DataWeave `toUpper(aString)` function converts a string to uppercase. Example: `toUpper("hello")` returns `HELLO`.

%dw 2.0

`toUpper(aString)`

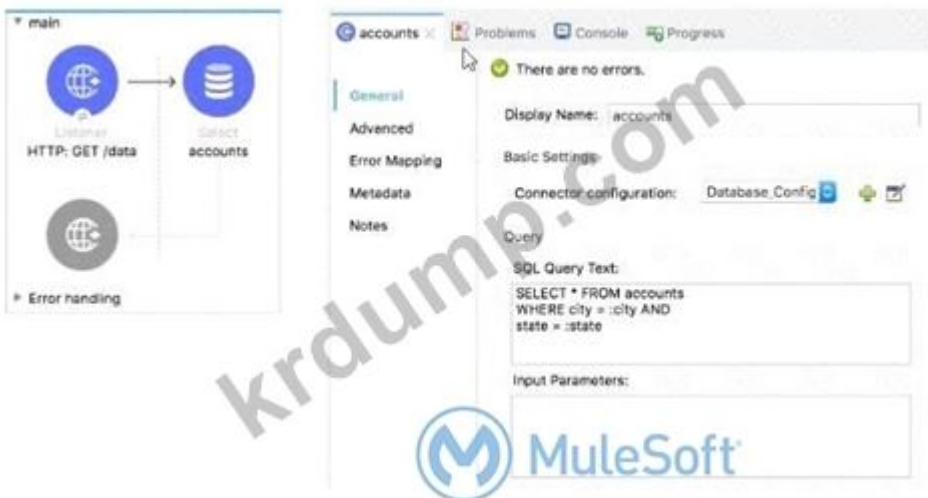
`toUpper(aString) = upper(aString)`

`toUpper("hello")`

MuleSoft link: <https://docs.mulesoft.com/mule-runtime/4.1/dataweave-functions>

NEW QUESTION: 9

Which of the following is a valid SQL query?



Which of the following is a valid 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. ☐☐ C
- B. ☐☐ D
- C. ☐☐ B
- D. ☐☐ A

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 10

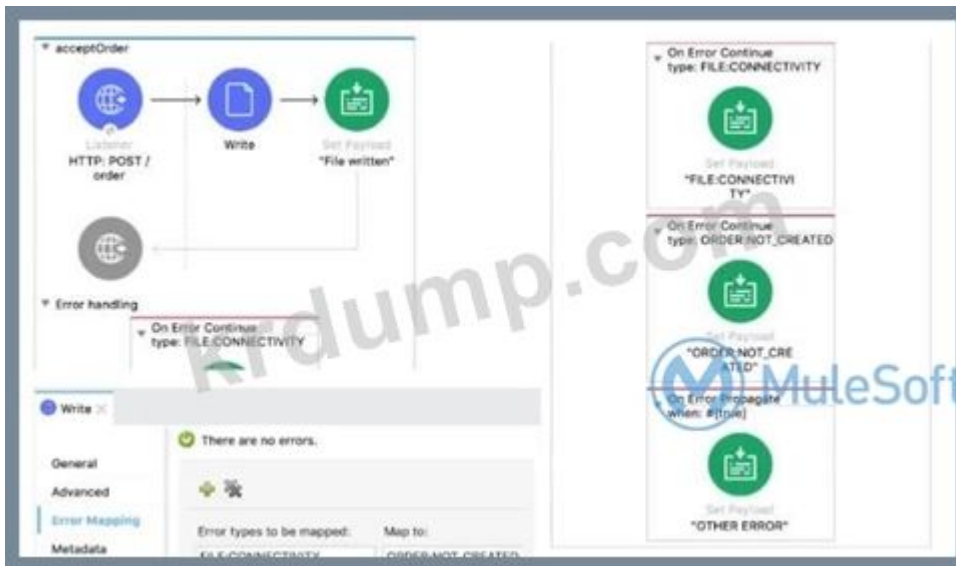
For Each ☐ Batch Job ☐☐☐☐☐ ☐☐ ☐☐ ☐☐☐☐☐☐☐?

- A. ☐ ☐ ☐☐ ☐☐☐☐☐☐.
- B. ☐☐ ☐☐☐ ☐☐ ☐☐☐☐☐☐☐ For Each☐ ☐☐ ☐☐☐☐☐☐.
- C. For Each☐ ☐☐ ☐☐☐☐☐☐☐ Batch Job☐ ☐☐ ☐☐☐☐☐☐☐.
- D. ☐ ☐ ☐☐ ☐☐☐☐☐☐☐.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 11

☐☐☐☐ ☐☐☐☐☐☐.



```
<flow name="acceptOrder">
  <http:listener doc:name="HTTP: POST /order" config-ref="HTTP_Listener_config"
    path="/order" allowedMethods="POST">
    <http:error-response >
      <http:body ><![CDATA[#[output text/plain -- payload]]></http:body>
    </http:error-response>
  </http:listener>
  <file:write doc:name="Write" config-ref="File_Config" path="newOrder.json">
    <error-mapping sourceType="FILE:CONNECTIVITY" targetType="ORDER:NOT_CREATED" />
    <file:content ><![CDATA[#[output appliation/json --- payload]]></file:content>
  </file:write>
  <set-payload value="#['File written']" doc:name=""File written"" />

```

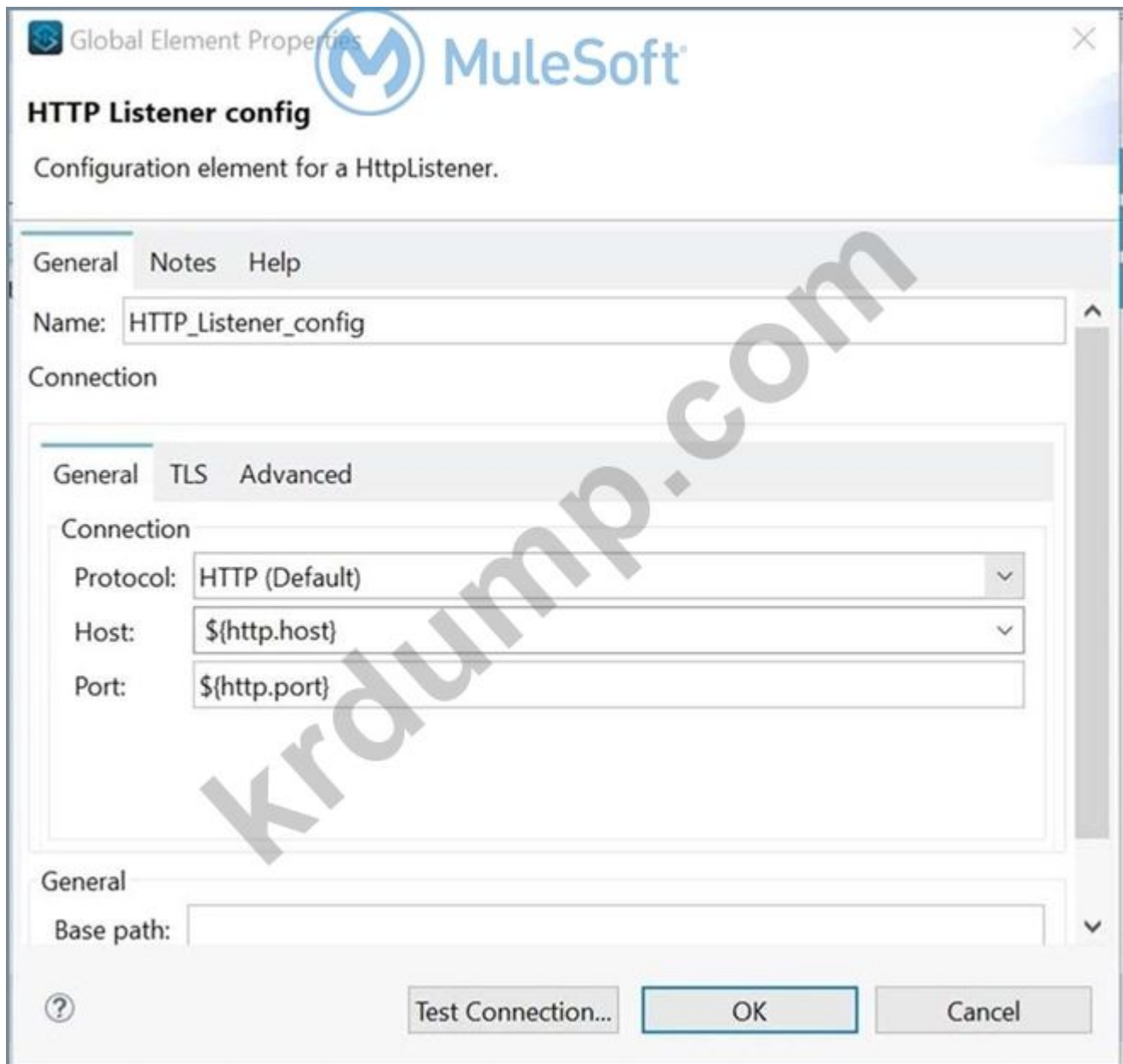
□ □□□□□ □□□□□ {"oid": "1000", "itemid": "AC200", "qty": "4"} □ POST □□□ Mule □ □□□□□ □□□□. □□ □□ □□□□ FILE:CONNECTIVITY □□□ □□□□□.

- A. "□□ □□"
- B. "□□□ □□"
- C. "□□: □□□□ □□"
- D. "□□:CONNECnvnY'

Answer: (SHOW ANSWER)

NEW QUESTION: 12

□□ □□□ □□□□□□. Mule □□□□□□□□ config.yaml□□□□ □□ □□ □□□□ □□□□ □□ □□ HTTP □□□□ □□ □□ □□ □□ □□□□ □□□□□□.



A. 1. http:

2. `␣␣␣ = "␣␣ ␣␣␣"`

3. `␣␣ = "8081"`

B. 1. http:

2. `␣␣ ␣␣: "api"`

3. `␣␣␣ : "␣␣ ␣␣␣"`

4. `␣␣ : "8081"`

(`␣␣`)

C. 1. `http.host = ␣␣ ␣␣␣`

2. `http.port = 8081`

D. 1. {

2. http:

3. `␣␣ ␣␣: "api",`

4. `␣␣: "8081",`

5. `␣␣␣: "␣␣ ␣␣␣"`

Answer: B (LEAVE A REPLY)

□□□ □□□ □□□□ □□□ □□□ □□□□.

http:

□□ □□: "api"

□□□ : "□□ □□□"

□□ : "8081"

NEW QUESTION: 13

□□□□ □□□□□.



```

<flow name="main" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP listener config" path="/" />
  <flow-ref doc:name="private" name="private"/>
  <set-payload value="Success - main flow" doc:name="Success - main flow" />
  <error-handler>
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - main flow" doc:name="Error - main flow" />
    </on-error-continue>
  </error-handler>
</flow>

<flow name="private" >
  <validation:is-number numberType="INTEGER" doc:name="payload" value="#[payload]"
  message="Validation Error" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - private flow" doc:name="Error - private flow" />
    </on-error-continue>
  </error-handler>
</flow>

```

□□ □□□ □□□ □□ □□ □□□□ □□□ □□□□□. □□ □□□□ HTTP □□□□ □□ □□□□□ □□□ □□ □□ □□□□ □□□□□?

- A. □□ - □□ □□
- B. □□□ □□ □□
- C. □□ - □□ □□
- D. □□ - □□ □□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 14

HTTP Listener □□□ APIkit □□□□ □□ □□□ □□□□□□ □□ □□ □□□ □□□□□□?

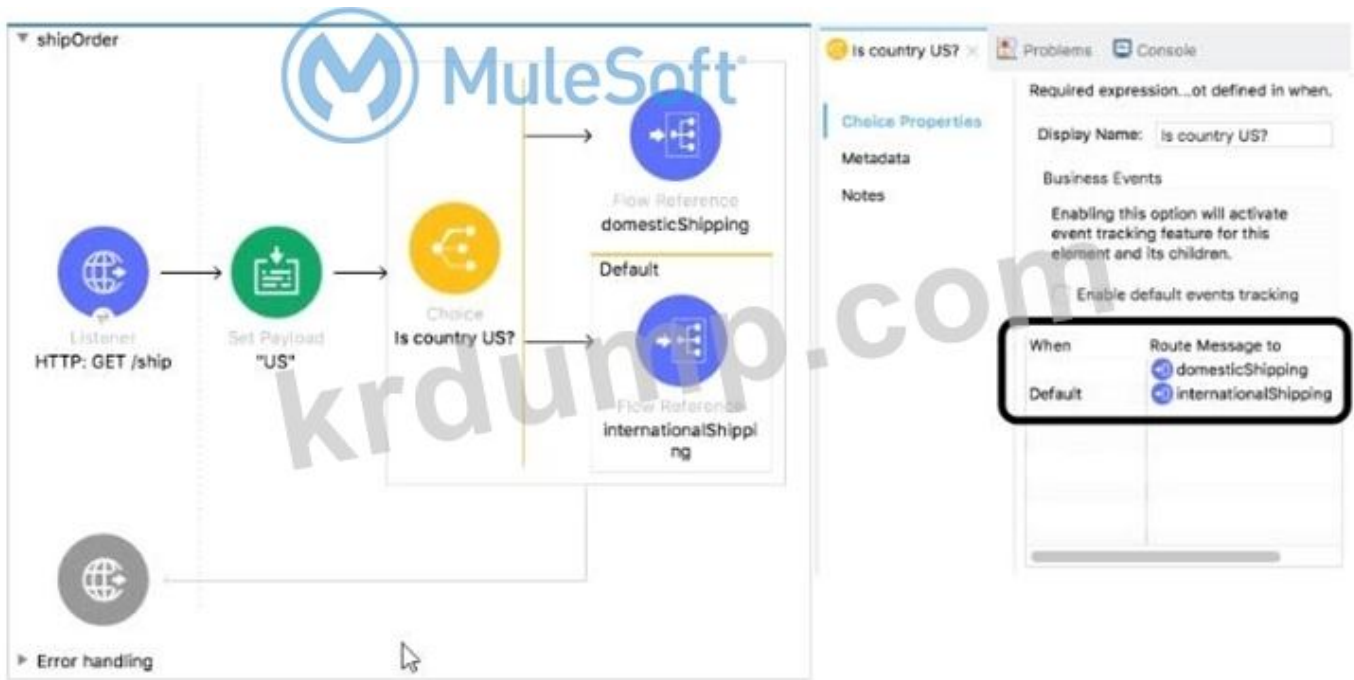
- A. /(*)
- B. /
- C. /()
- D. "/*"

Answer: (SHOW ANSWER)

Option1 □ HTTP Listener □□□ □□□□ □□ □□□ □□□□□□.

NEW QUESTION: 15

□□□ □□□□□.



□□□□ documenticShipping □□□□ □□□□□ □□ Choice □□□□ when □□□□ □□ □□□ □□□□ □□□□□?

- A. 0#[□□□□ = '□□']
- B. #[□□□□ == '□□']
- C. #[if(□□□□ = '□□')]
- D. #[if(□□□□ == "□□")]

Answer: (SHOW ANSWER)

□□ 1□ □□□ □□□□□.

NEW QUESTION: 16

□□□ □□□□□.



□□□□ □□ □□□ □□ □□, □□ □ □□□□ □□ □□□ □□ □□□□□□.
 □□ □□□ □□□ □□□ □ □□□□ □□□ □□□ □□□□□ □□□□□□?

A)

```

[
  {
    "attributes": ...,
    "payload": "100"
  },
  {
    "attributes": ...,
    "payload": "200"
  }
]

```

B)



```
{
  "0": "100",
  "1": "200"
}
```

C)

```
["100", "200"]
```

D)

```
{
  "0": {
    "attributes": "...",
    "payload": "100"
  },
  "1": {
    "attributes": "...",
    "payload": "200"
  }
}
```

A. C

B. A

C. B

D. D

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 20

□□□□ □□□□□. □□□□ API□□ □□□ □ Book □□□ □□□ Book □□□ □□□□□
□. □ □ □□□ □□□ □ □□□ □□□□ API□ □□□ RAML□ □□□□□?

```

#%RAML 1.0 DataType
# bookDataType.raml

type: object
properties:
  ID?: integer
  title: string
  author: string
  publisher?: string
  year: integer
  ISBN:
    type: string
    required: true

#%RAML 1.0 NamedExample
# bookExample.raml

bookExample:
  ID: 101
  title: Shakespeare
  author: Encyclopaedia Britannica
  publisher: John Wiley & Sons
  year: 2007
  ISBN: "0471767840"

```

A)

```

#%RAML 1.0
title: Books

Book: !include BookDataType.raml

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

```

B)

```

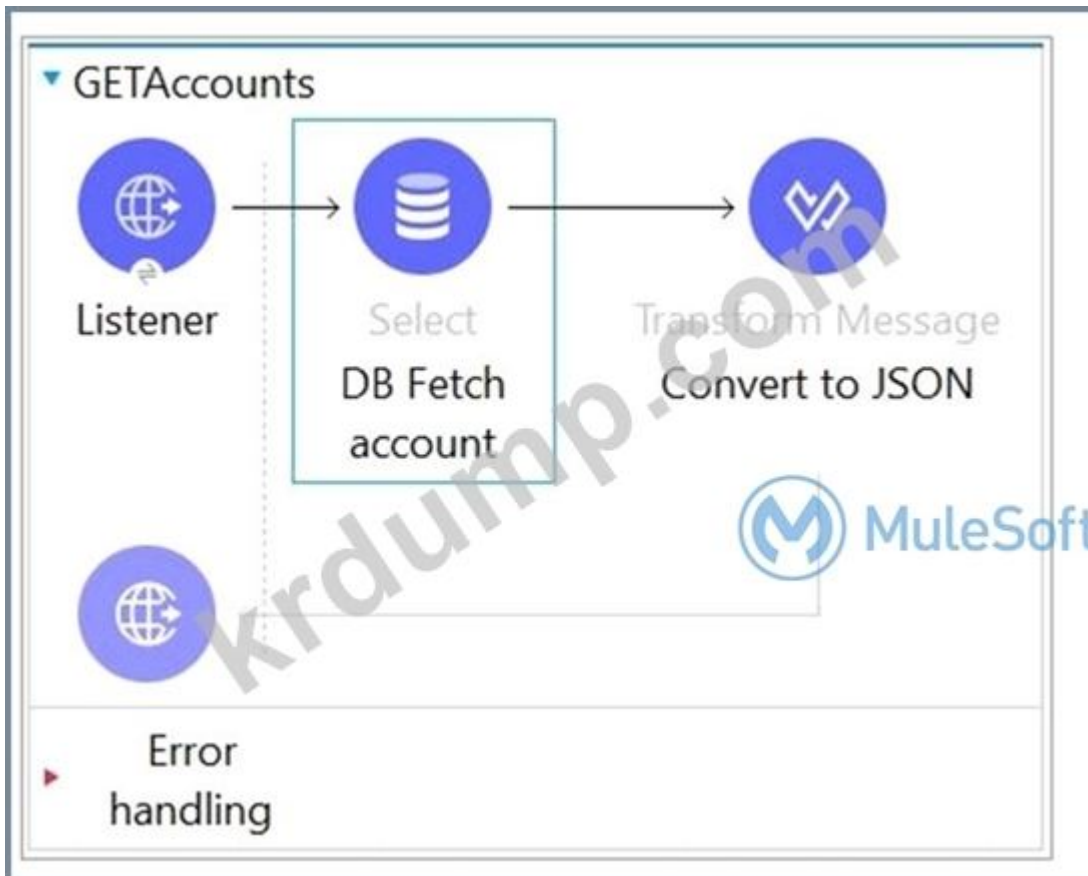
#%RAML 1.0
title: Books

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)



Display Name: DB Fetch account

Basic Settings
Connector configuration: Database_Config

Query
SQL Query Text:

Input Parameters: `1={
2= 'city': attributes.queryParams.city,
3 'state': attributes.queryParams.state
4 }`

SQL `SELECT * FROM accounts WHERE city = :city AND state = :state`

- A. WHERE `city = :city` AND `state = :state`
- B. `city = attributes.city` AND `state = attributes.state`
- C. WHERE `city = :city` AND `state = :state`
- D. WHERE `city = :city` AND `state = :state`

Answer: C (LEAVE A REPLY)

where `city = :city` AND `state = :state`

NEW QUESTION: 22

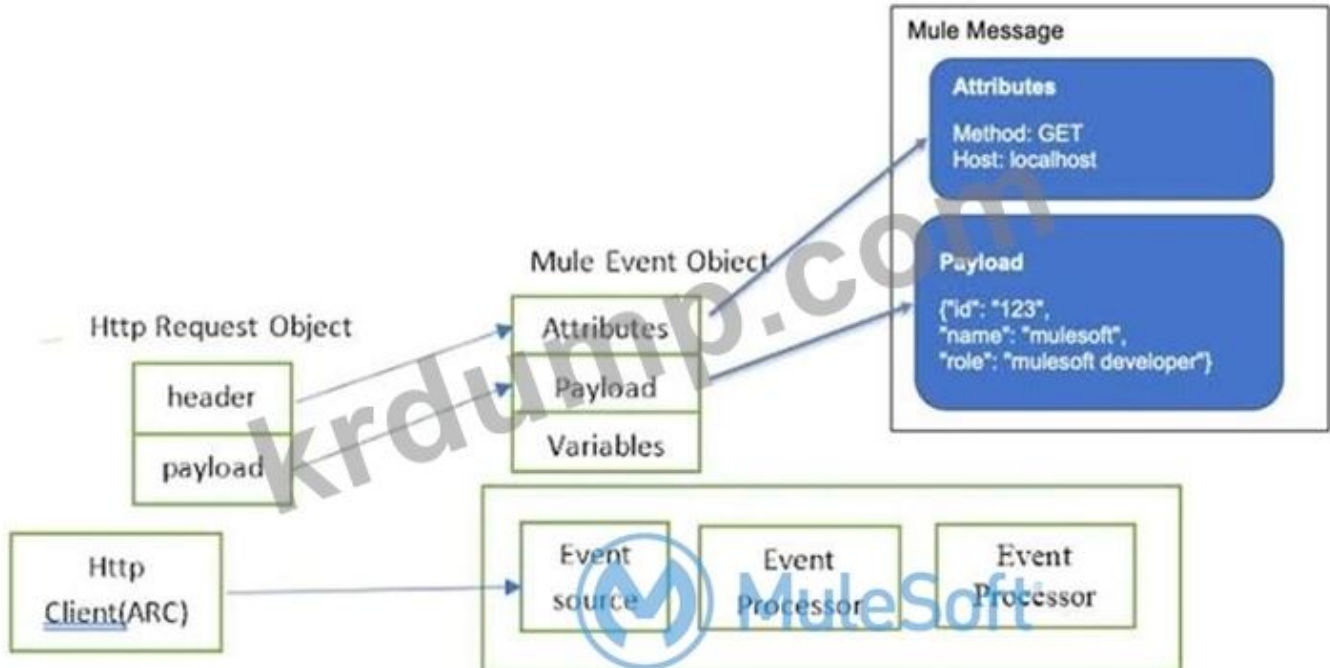
HTTP Listener `SELECT * FROM accounts` Mule `SELECT * FROM accounts`

- A. □□□□ □□
- B. □□
- C. □□
- D. □□□□

Answer: (SHOW ANSWER)

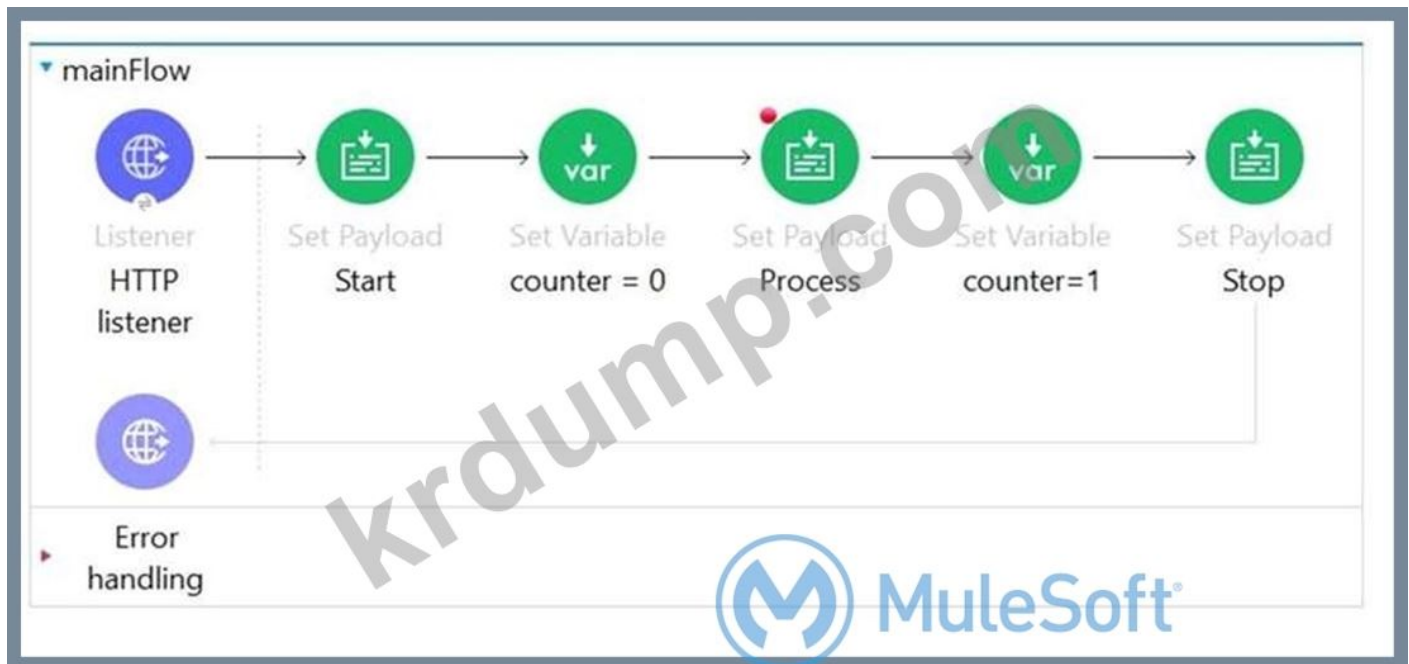
□□□ □□□□□.

□□ □□□□, URI □□□□ □ □□□ □□□ □□□ □ □□ □□□□.



NEW QUESTION: 23

□□□□ □□□□□.



□ □□□□□□□ Anypoint Studio□□ □□□□□ □□ □□□ □□ □□□□□ □□□□□.
 □ □□□□□□ □□□□ □□□□□ □□□□□ □□ □□□□□□?

- A. □□

B. □□□□

C. □□

D. □□□□□ □□□□□ □□ □□ □□□□.

Answer: A ([LEAVE A REPLY](#))

□□□ □□□ □□□ □□□□□□□ □□□□□.

NEW QUESTION: 24

□□□ □□□□□.



Mule □□□□ □□□ □□ □□□ □□□□□. □□ □□□□ □□□ □□□ □□□□□?

A. □ □□□

B. □ □□□ □□□□

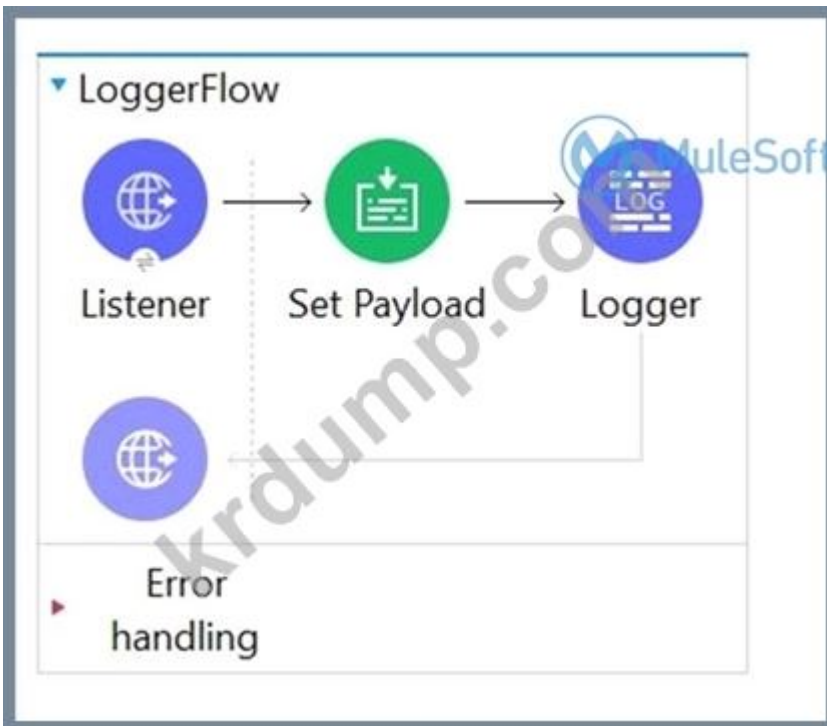
C. □ □□□

D. □ □□□ □□

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 25

□□□□□ □□□□□□.



```

<flow name="LoggerFlow" doc:id="d5015e61-b3b5-4833-8c5e-ed176a3f6cb0" >
  <http:listener doc:name="Listener" doc:id="5ae3d668-7075-4236-a523-a08b03186b53" config-ref="HTTP_listener_config" path="/Log" />
  <set-payload value="#[{
$#10;   "student": {
$#10;     "name": "Anay",
$#10;     "age": 6
$#10;   }
$#10;}]" doc:name="Set Payload" doc:id="7763301e-1fed-40fc-968d-47c1b113c867" />
  <logger level="INFO" doc:name="Logger" doc:id="8e1c416b-78bd-44fb-b0db-cd5b3d382c6d" message='Result [{"INFO":'+ payload}]' />
</flow>

```

□□□□ □□ □□□□ □□□□□ □□□ □□□□□. □□ □□ □□□ □□□ □□□ "□□ #["INFO"++ □□□□]" □□□□ □□□□□. □ □□□ □□□ □ □□□ □□□ □□□□□?

- A. □□ INFO□□□□
- B. □□ □□{"□□":{"□□":"□□□","□□":6}}
- C. 1. 1. "□□ □□□ □□□□ '++' □□□ □□□□□□□.
- 2. 2. 1: □□□("□□")
- 3. 3: □□({□□: {□□: "Anay" as String {class: "java.lang.String"}, age: 6 as Numbe...})
- D. □□: ++ □□ □□□ □□□ #□ □□□□□□.

Answer: C (LEAVE A REPLY)

□□ □□□ □□□ □□ □□□□□□ □□□□□ □□□ □□□ □□□□□. □ □□ □□□□□ □□□□□.

- "□□ □□□ □□□□ '++' □□□ □□□□□□□.
- 1: □□□("□□")
- 2: □□({□□: {□□: "Anay" as String {class: "java.lang.String"}, □□

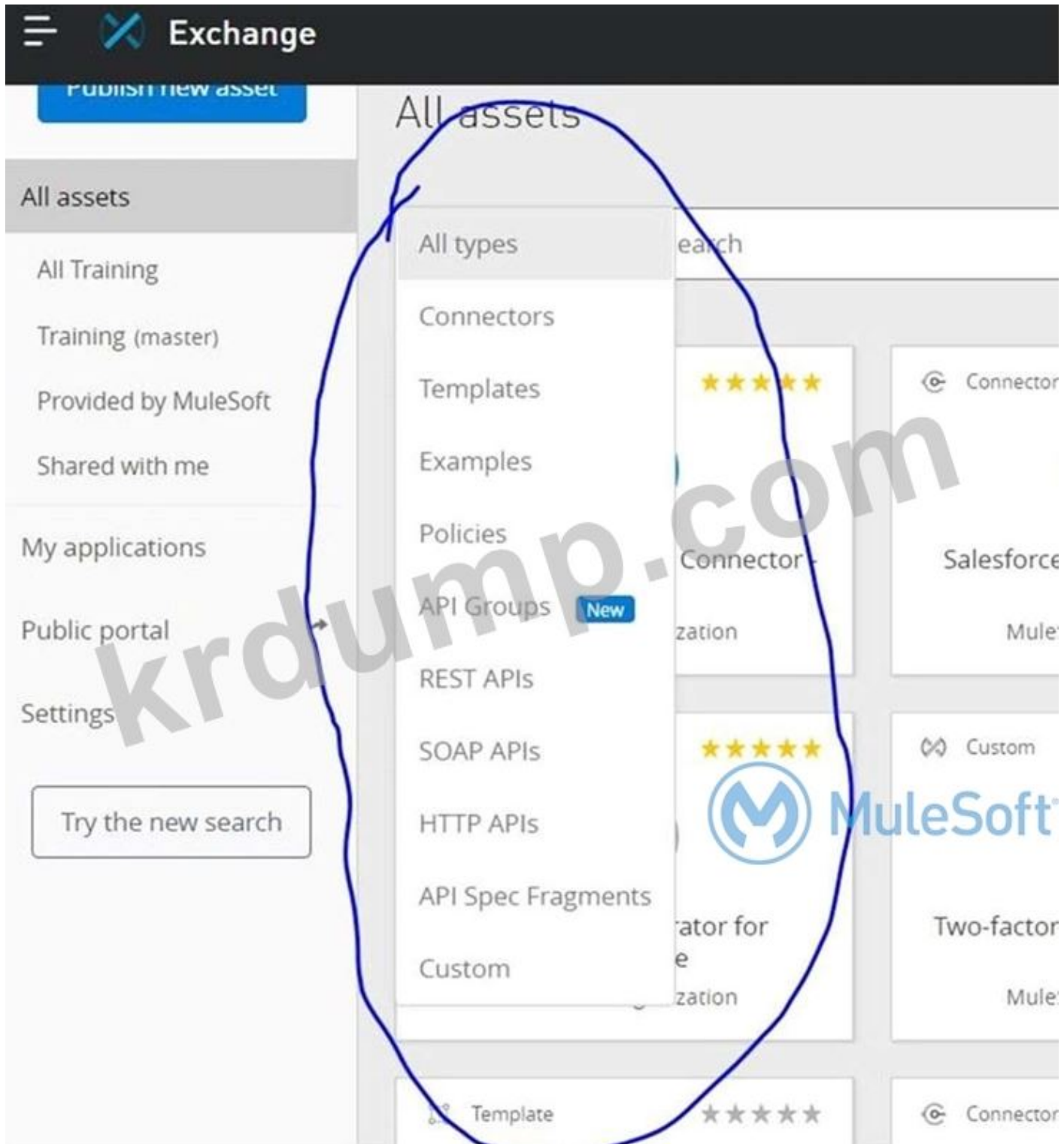
NEW QUESTION: 26

□□ □ □□□ □□ □□?

- A. □□□
- B. □□□
- C. □□
- D. □

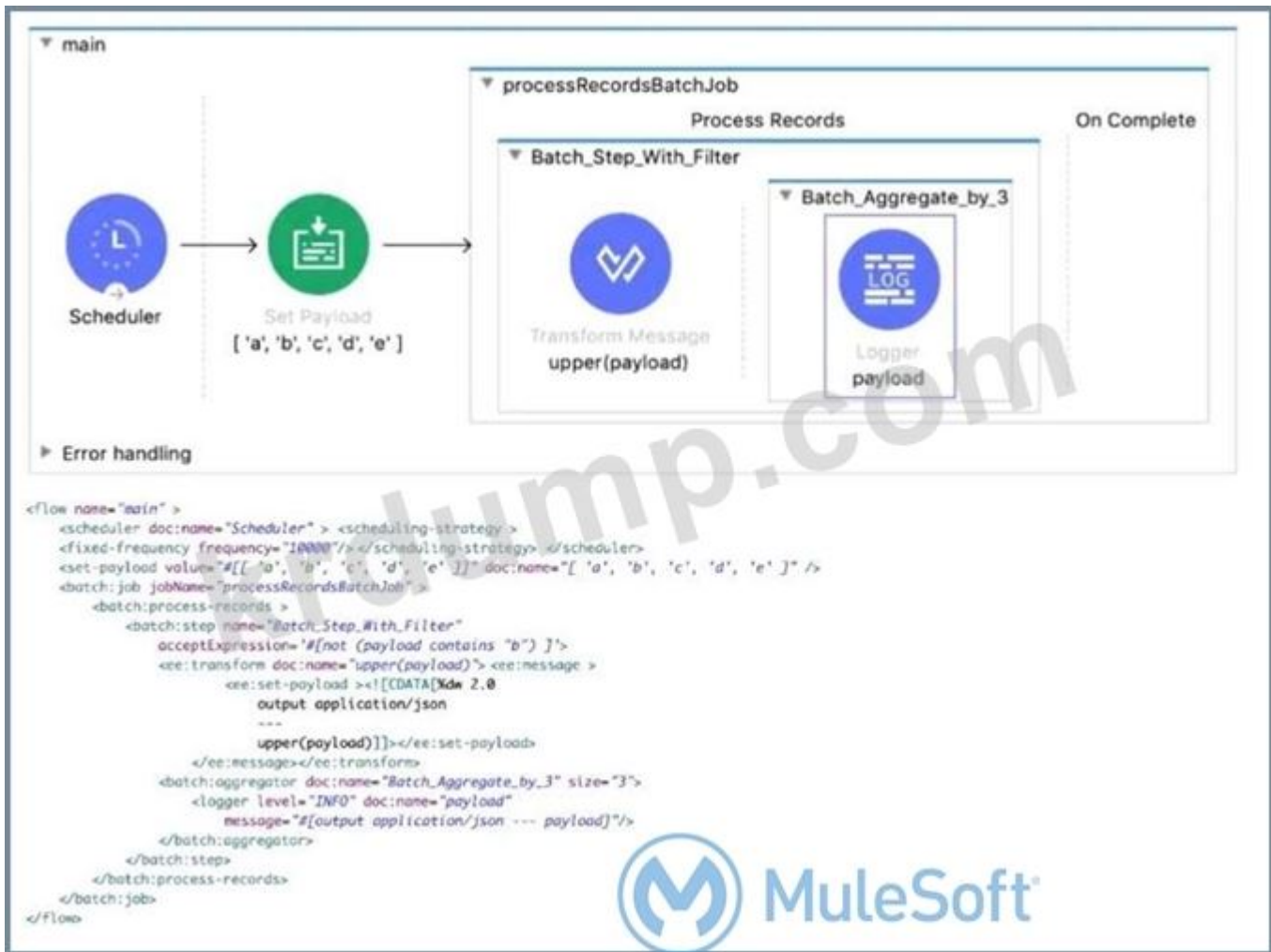
Answer: C (LEAVE A REPLY)

□□□ □□□□ □□□ □□□□□. □□□□ □□ □□ □□□□□.



NEW QUESTION: 27

□□□□ □□□□□.



□□□□ □□ □□□ □□ □□, □□□ □ □□□□ □□ □□□ □□ □□□□□□. □□ □□□ □□□ □□□ □ Logger □□ □□□ □□□ □□□ □□□□□□□□□□□□?

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

Answer: (SHOW ANSWER)

□□□ □□□ □□□□.

□□ 2021-06-09 19:14:56,039 [[MuleRuntime].uber.06: [validationtest].batch-job-validationtestBatch_Job-work-manager @6de10f3e] [□□□□: validationtestFlow/processors/1/route/0/route /0/□□□□□□/□□□□□/0; □□□: bfb751e1-9939-11eb-9f69-02053763653a]

org.mule.runtime.core.internal.processor.LoggerMessageProcessor:

```

[
  "\□\\"",
  "\□\\"",
  "\□\\""
]

```

--

2021-06-09 19:15:02,486 [[MuleRuntime].uber.06: [validationtest].batch-job-validationtestBatch_Job-work-manager @6de10f3e] [validationtestFlow/processors/1/route/0/route /0/bfb751e1-9939-11eb-9f69-02053763653a] org.mule.runtime.core.internal.processor.LoggerMessageProcessor: ["\"]] ["A", "C", "D"] ["E"]

[""]

NEW QUESTION: 28

Cloudhub Mule [?] [?]

- A. [?]
- B. [?]
- C. Mule [?]
- D. [?] 1 [?] 2 [?]

Answer: A (LEAVE A REPLY)

[?] [?] Mule [?] [?] Mule [?]

NEW QUESTION: 29

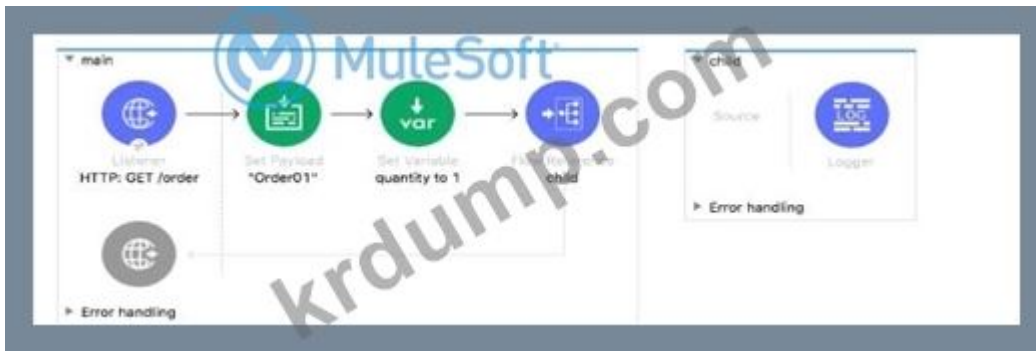
[?] [?] Scatter_Gather [?] flow1 [?] 10 [?] flow2 [?] 20 [?] Scatter_Gather [?] [?]

- A. 20
- B. 30
- C. 0
- D. 10

Answer: A (LEAVE A REPLY)

NEW QUESTION: 30

[?] [?]



□□ □□□□ □□ □□□ □□ □□ □□□ □□□□□.

□ □□□□□□ http://localhost:8Q81/order□ □□□ □□□ □ □□ □□□□ □□□□ □ □□ □□ □□□□□? □□=□□□?

A. □□□□

B. □□□□

□□ □□

C. □□□□

□□ □□ □□□□

D. □□□□

□□ var □□ □□ □□□□

Answer: D (LEAVE A REPLY)

□□□

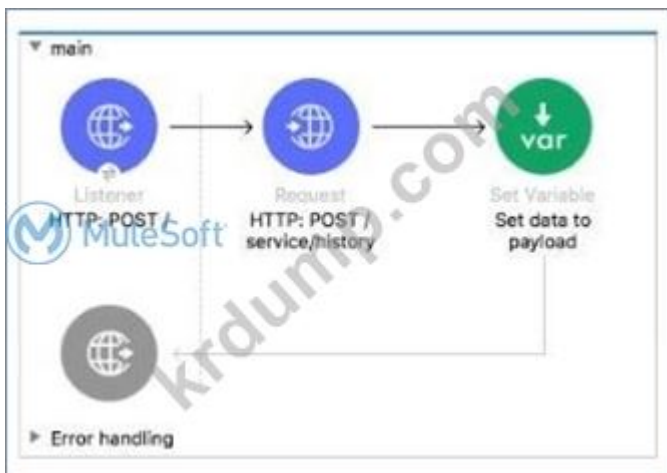
□□ □□□

□□ □□

□□ □□ □□□□

NEW QUESTION: 31

□□□ □□□□□.



□□ □□ □□ □□ □□□□ □□□□ □□ □□□ □□□ □□□ □□□ □□□□□?

A. Mule □□□ □□ □□□ □□

B. □□ □□

C. Mule □□□ □ □□□ □□

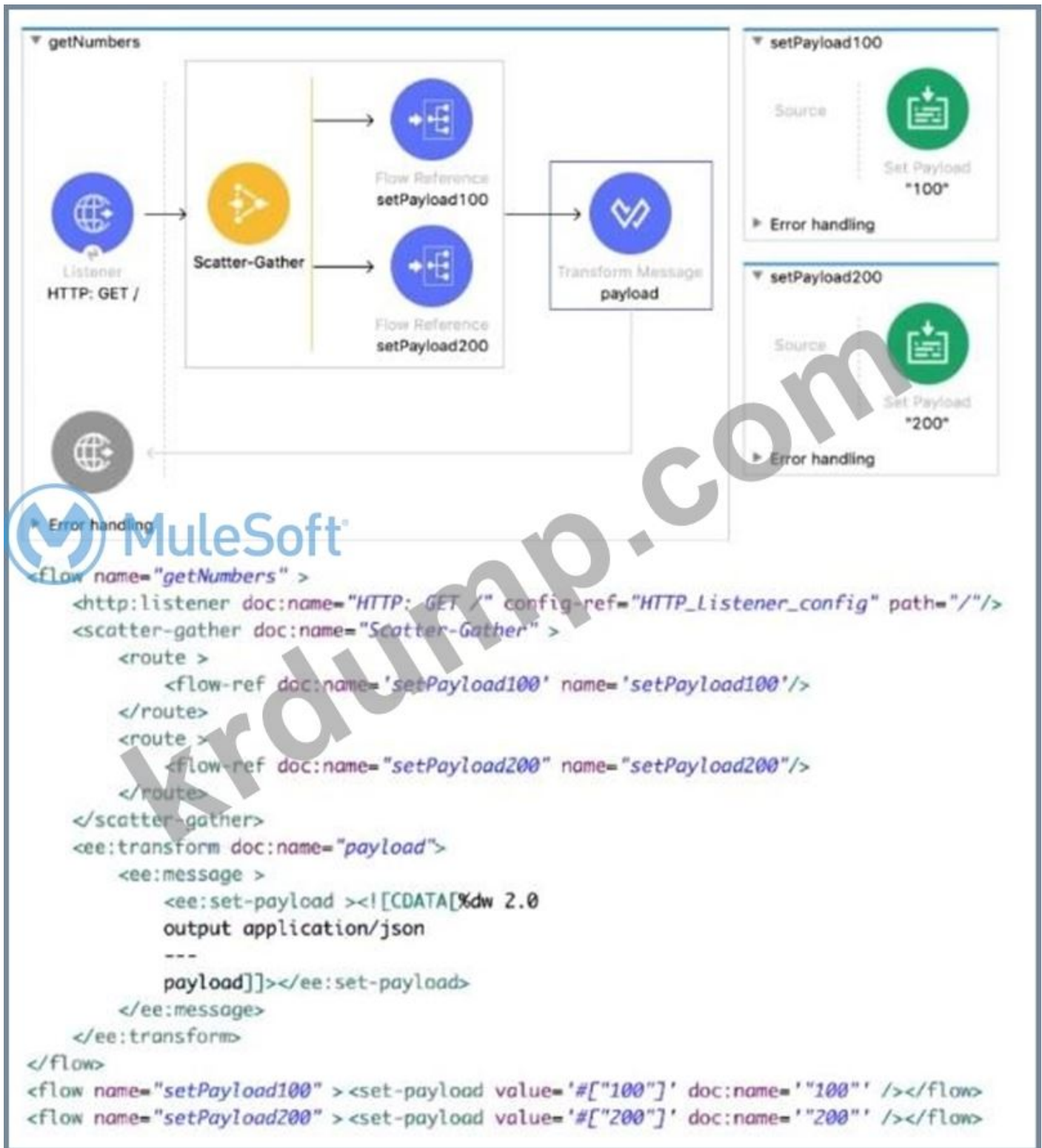
D. ObjectStore□ □/□ □

Answer: (SHOW ANSWER)

MCD-Level-1 ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ MCD-Level-1 ☐
☐! DumpTop ☐ ☐☐ MCD-Level-1 ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop MCD-Level-1 ☐☐
☐☐☐ ☐☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐. ☐☐☐☐☐ ☐☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop
MCD-Level-1 ☐☐☐ ☐☐☐☐☐. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
(235 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 32

☐☐☐☐ ☐☐☐☐☐.



□□□□ □□ □□□ □□ □□, □□ □ □□□□ □□ □□□ □□ □□□□□.

□□ □□□ □□□ □□□ □ Logger □□ □□□ □□□ □□□□ □□□□□□?

A)

```
[
  {
    "attributes": ...,
    "payload": "100"
  },
  {
    "attributes": ...,
    "payload": "200"
  }
]
```

B)

```
{
  "0": "100",
  "1": "200"
}
```

C)

```
["100", "200"]
```

D)

```
{
  "0": {
    "attributes": ...,
    "payload": "100"
  },
  "1": {
    "attributes": ...,
    "payload": "200"
  }
}
```

A. A

B. D

C. C

D. B

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 33

.

```
##%RAML 1.0
title: Accounts API
version: 1.0

/accounts:
  get:
    description: Get all accounts
    responses:
      200:
        body:
          application/json:
            example:
              id: "48292"
              name: Geordi La Forge
              address: 1 Forge Way, Midgard, CA 95928
              customer_since: "2014-01-04"
              balance: 4829.29
  post:
    description: Create an account
    body:
      application/json:
        example:
          name: Geordi La Forge
          address: 1 Forge Way, Midgard, CA 95928
          customer_since: "2014-01-04"
```

POST /accounts □□□□□□□ □□□□ □□□□ □□□□□?

A)

```
{
  "id": "48292",
  "name": "Geordi La Forge",
  "address": "1 Forge Way, Midgard, CA 95928",
  "customer_since": "2014-01-04",
  "balance": 4829.29
}
```

B)

```
<item>
  <id>48292</id>
  <name>Geordi La Forge</name>
  <address>1 Forge Way, Midgard, CA 95928</address>
  <customer_since>2014-01-04</customer_since>
  <balance>4829.29</balance>
</item>
```

C)

```
<item>
  <name>Geordi La Forge</name>
  <address>1 Forge Way, Midgard, CA 95928</address>
  <customer_since>2014-01-04</customer_since>
  <balance>4829.29</balance>
  <bank_agent_id>48-SJT-282924-KL</bank_agent_id>
</item>
```

D)

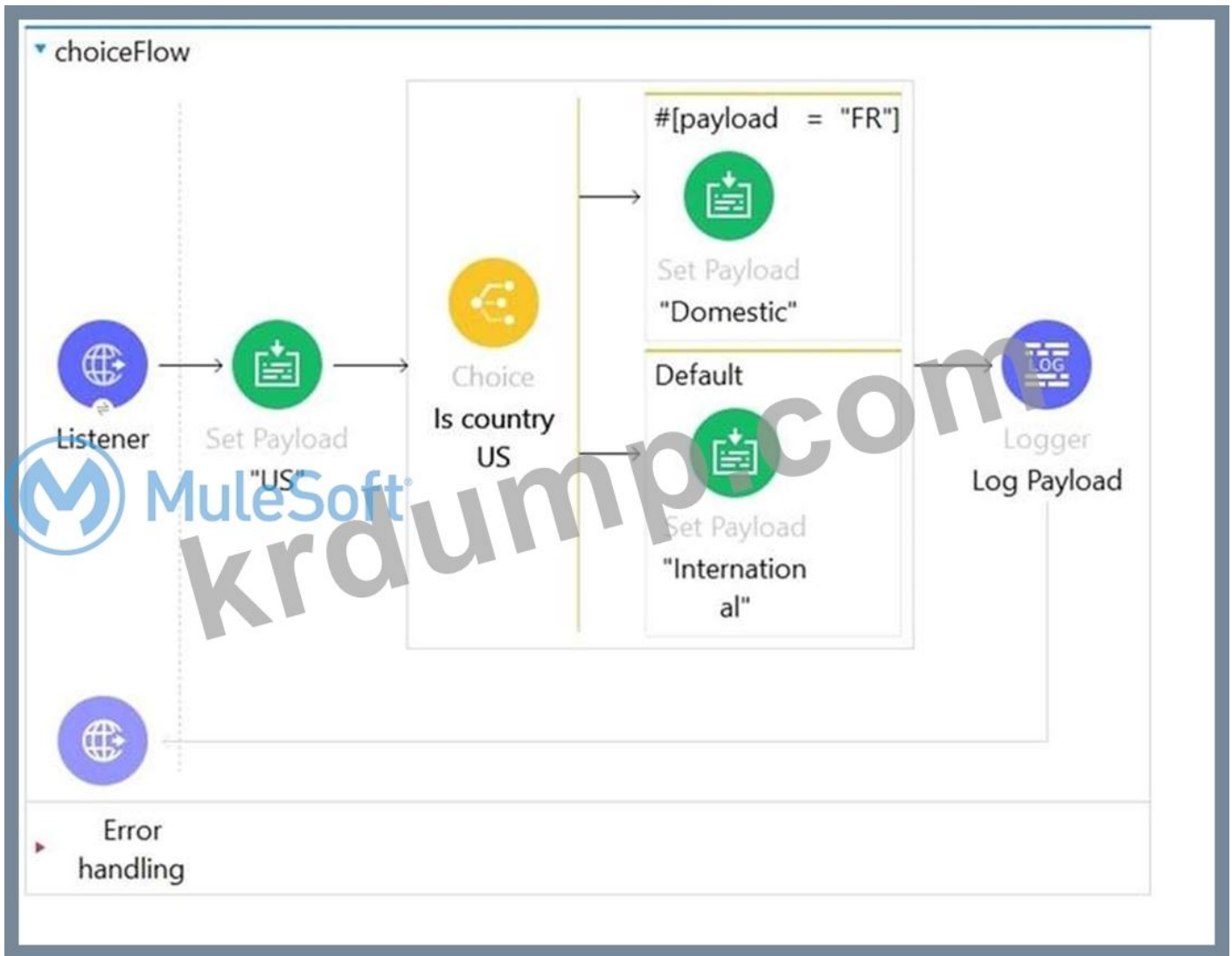
```
{
  "name": "Geordi La Forge",
  "address": "1 Forge Way, Midgard, CA 95928",
  "customer_since": "2014-01-04",
  "balance": 4829.29
  "bank_agent_id": "48-SJT-282924-KL"
}
```

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

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 34

□□□□ □□□□□. □□ □□□□□ □□ □□ □□□ □□ When □□□□ "#[payload=
"FR"]"□ □□□□□.
□□ □□□ □□□ □□□ □ □□□ □□□ □□□□□?



A. "□□"

*

B. "□□"

*

C. "□□"

*

D. □□□ □□ □□ □□

(□□)

Answer: D (LEAVE A REPLY)

□□□□□ DataWeave □□□ #[payload == "FR"]□□□. □ □□ □□□ =□ □□□□□ □□ □ □□□□□.

NEW QUESTION: 35

□ □□ □□ □□□□, itemID□ □□ □□ □□ productCategory□ □□ □□□ □□ □□□□ □ □□ □□□ □□□□ newProdCode□□ □□□ □□□□ □□□. newProdCode □□□ □□□□ □□□□ DataWeave □□□ □□□□□?

A. var newProdCode(itemID: □□, productCategory: □□□) -> "PC-" ++ productCategory ++(□□ ID□ □□□□)

B. function newProdCode(itemID: □□, productCategory: □□□)

"PC-" ++ productCategory ++(□□ ID□ □□□□)

C. fun newProdCode{itemID: □□, productCategory: □□□} -> "PC-" ++ productCategory ++(□□ ID□ □□□□)

D. fun newProdCode(itemID: □□, productCategory: □□□) = "PC-" ++ productCategory ++(□□ ID□ □□□□)

Answer: (SHOW ANSWER)

NEW QUESTION: 36

□□□ □□ □□ □ □□□ □□□ □□□□ □□□□ □□□□ □□□□. □□□□ HTTP □□ □□ □□□□ □□□□□□ SELECT □□□ □□□□□.

□ □□ □□□ □□□□□ □ □□ □□□ □□□□□ □□□□ □□□ □ □□□□□ □□ □□ □□ □□□□□?

- A. □□□□□□ SELECT □□□ □□ □□□ □□□□.
- B. Database SELECT □□□ Message Enricher □□□ □□□□.
- C. □□, □□ □□□□□ □□ □□□□□ □□□□□.
- D. Database SELECT □□□ □□□□□ □□□ □□

Answer: D (LEAVE A REPLY)

□□□ □□□□□□ SELECT □□□ □□□□□ □□□ □□□□□. HTTP □□□ □□□ □□ □□□ □□□□□ □□□□□□ SELECT□ □□□ □□□ □ □□□□. □□□□ □□ □□ □□ □ TransformMessage□ □□□□ □ □□ □□□ □□□□□ □□□ □□□□ □□□□.

NEW QUESTION: 37

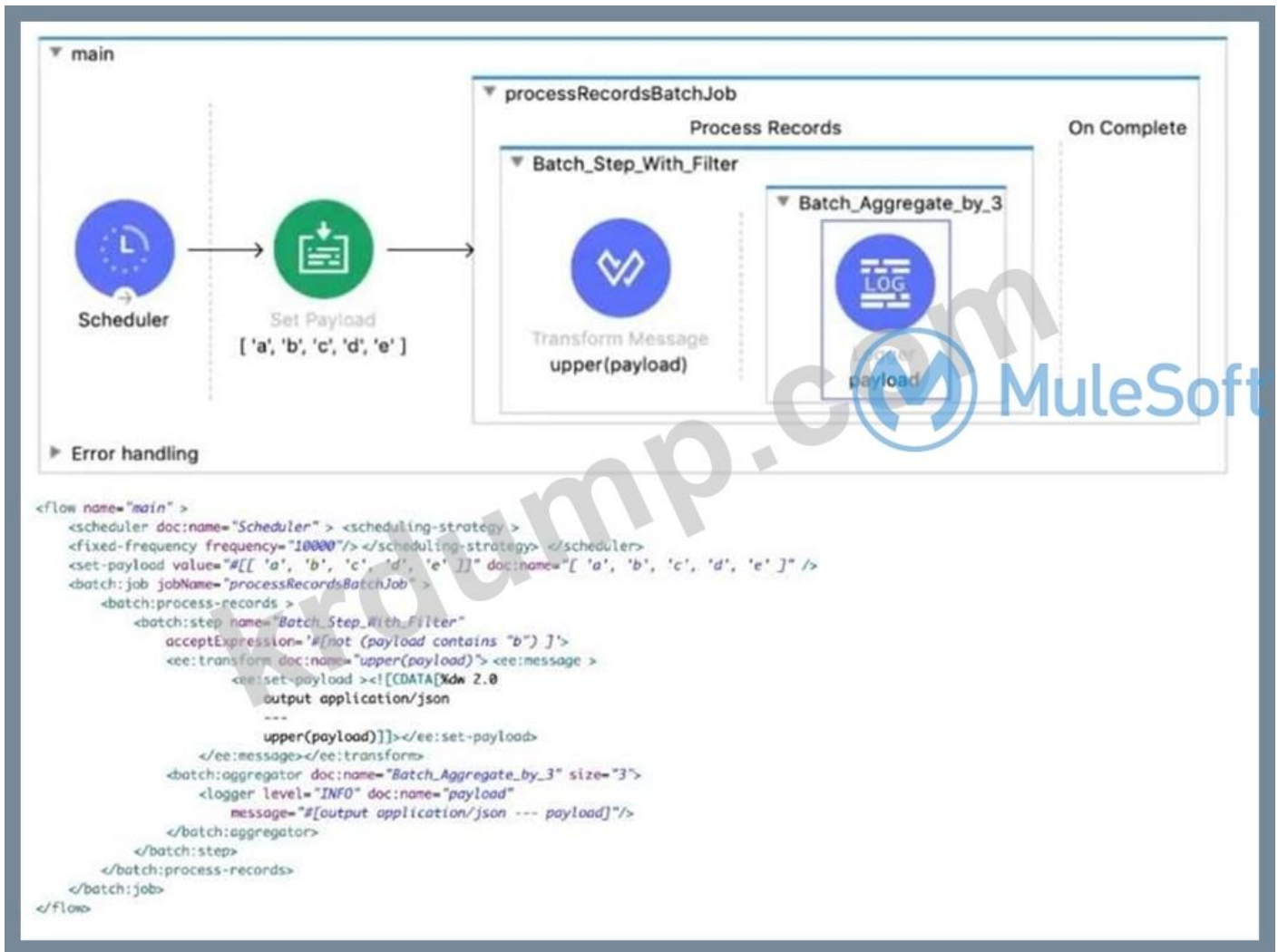
API □□□ RAML□ □□□□ □□□□□□□□. □ API □□□□ REST □□□□ □□□ □□ □□ □□□□□?

- A. Mule □□□□□ src/main/resources/api □□□ □□ □□
- B. API □□□ Any point Exchange□ □□
- C. Design Center□□ Flow Designer□ □□□□ API □□ □□
- D. API □□ □□□□ □ APIkit□ □□□□ □□□□□ □□

Answer: B (LEAVE A REPLY)

NEW QUESTION: 38

□□□□ □□□□□.



Which of the following is the correct payload after the Batch_Aggregate_by_3 step? (Note: The payload is a list of lists, where each inner list contains the characters from the previous step that were not filtered out.)

- A. ["D", "E"]
- B. [{"A", "C", "D"}, {"E"}]
- C. ["E"]
- D. ["A", "C", "D", "E"]

Answer: (SHOW ANSWER)

NEW QUESTION: 39

Utility.dwl is located in src/main/resources/modules. Utility.dwl is used to pascalize the DataWeave code. Which of the following is the correct way to use the pascalize function?

A)

```

%dw 2.0
output application/json
import modules Utility
pascalize( "max mule" )

```

B)

```
tdw 2.0
output application/json
import modules::Utility
---
pascalize( "max mule" )
```

C)

```
tdw 2.0
output application/json
import modules::Utility
---
Utility::pascalize( "max mule" )
```

D)

```
tdw 2.0
output application/json
import modules::Utility
---
Utility.pascalize( "max mule" )
```

A. C

B. B

C. A

D. D

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 40

.

```
##RAML 1.0
title: ACME Telecom API
version: 1.0

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            example: |
              [
                {
                  "plan_type": "Super Saver 500",
                  "plan_details": "all-inclusive",
                  "monthly_discount": 0.10
                },
                {
                  "plan_type": "Business Plus 1000",
                  "plan_details": "business package",
                  "monthly_discount": 0.20
                }
              ]
```

API object in Anypoint Exchange. ACME/DataTypes/PlanDataType.raml. RAML format?

A)

```
##%RAML 1.0
title: ACME Telecom API
version: 1.0

dataTypes:
  Plan: !include ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !include ACME/Examples/PlanExamples.raml
```

B)

```
##%RAML 1.0
title: ACME Telecom API
version: 1.0

Types:
  Plan: !reference ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !reference ACME/Examples/PlanExamples.raml
```

C)

```
##%RAML 1.0
title: ACME Telecom API
version: 1.0

dataTypes:
  Plan: !reference ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !reference ACME/Examples/PlanExamples.raml
```

D)

```

#%RAML 1.0
title: ACME Telecom API
version: 1.0

types:
  Plan: !include ACME/DataTypes/PlanDataType.raml

/plans:
  get:
    responses:
      200:
        body:
          application/json:
            type: Plan[]
            example: !include ACME/Examples/PlanExamples.raml

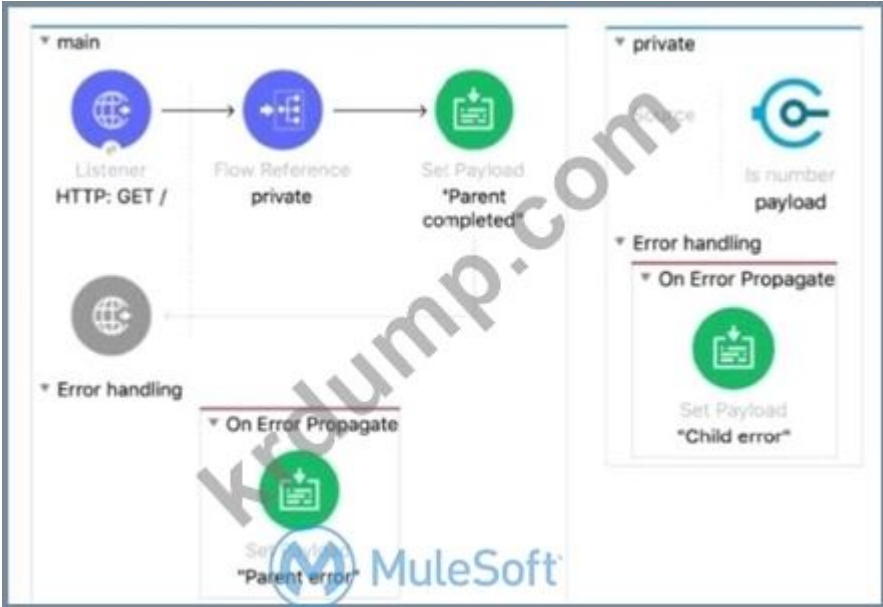
```

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 41

.



Mule .

.

HTTP ?

```

http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener config" >
  <http:listener-connection host="0.0.0.0" port="8081" />
</http:listener-config>

flow name="main" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <flow-ref doc:name="private" name="private"/>
  <set-payload value="Parent completed" doc:name="Parent completed" />
  <error-handler>
    <on-error-propagate enableNotifications="true" logException="true" doc:name="On Error Propagate" >
      <set-payload value="Parent error" doc:name="Parent error" />
    </on-error-propagate>
  </error-handler>
</flow>

flow name="private" >
  <validation:is-number numberType="INTEGER" doc:name="payload" value="#[payload]"
  message="Validation Error" />
  <error-handler >
    <on-error-propagate enableNotifications="true" logException="true" doc:name="On Error Propagate" >
      <set-payload value="Child error" doc:name="Child error" />
    </on-error-propagate>
  </error-handler>
...

```

- A. "[]"
- B. "[]"
- C. "[]"
- D. "[]"

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 42

□□□□ □□□□□.



```

1 %dw 2.0
2 output application/json
3 ---
4 order: {
5   item: {
6     itemName: payload.item,
7     itemType: payload.itemType,
8     price: payload.price
9   }
10 }

```

- □□ □□□ addItem □□ □□□□ DataWeave □ □□□□ □□ □□□ □□□□□.
- addItem □□ □□□ □□□□ □□□ 100□ □□□ □□□□ □□□ □□□□ □□ createOrder
- Set Payload □□□□ □□ □□□ DataWeave □□□ □□□□□?
- A. lookupf "addItem", { □□□□: { price: "100", item: "router", itemType: "cable" } })
 - B. addItemf { □□□□: { price: "100", item: "router", itemType: "cable" } >)
 - C. addItemf { price: "100", item: "router", itemType: "cable" } }
 - D. lookupf "addItem", { price: "100", item: "router", itemType: "cable" } })

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 43

□□□□ □□□□□.

```

<flow name="main">
  <http:listener doc:name="HTTP: POST /jms" config-ref="HTTP_Listener_Config" path="/jms" />
  <jms:publish doc:name="JMS: one" config-ref="JMS_Config" destination="one">
    <jms:message outboundContentType="text/plain" />
  </jms:publish>
  <http:request method="POST" doc:name="HTTP: POST /data" url="http://localhost:8081/data"/>
  <jms:publish-consume doc:name="JMS: two" config-ref="JMS_Config" destination="two">
    <jms:message outboundContentType="text/plain" />
  </jms:publish-consume>
  <set-payload value="#[payload ++ 'Three']" doc:name='payload as String ++ "Three"' />
</flow>
  
```

□ □□□□□□ "Hello-" □□□□□ □□ HTTP □□□□□ POST □□□ □□□□□. □ □□□□□□
 □ □□ □□□ □□□□□□?
 □ □□□□□□ □□ □□□ □□□□□□?

- A. Helb-JMS1-HTTP-JMS2 -3
- B. HTTP-JMS2-3
- C. Hello- HTTP-] MS2-□□
- D. Hello-HTTP-Three

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 44

□□□□ □□□□□.



```
<flow name="acceptOrder">
  <http:listener doc:name="HTTP: POST /order" config-ref="HTTP_Listener_config"
    path="/order" allowedMethods="POST">
    <http:error-response >
      <http:body ><![CDATA[#[output text/plain --- payload]]]></http:body>
    </http:error-response>
  </http:listener>
  <file:write doc:name="Write" config-ref="File_Config" path="newOrder.json">
    <error-mapping sourceType="FILE:CONNECTIVITY" targetType="ORDER:NOT_CREATED" />
    <file:content ><![CDATA[#[output application/json --- payload]]]></file:content>
  </file:write>
  <set-payload value='#[ "File written" ]' doc:name="File written" />
</flow>
```

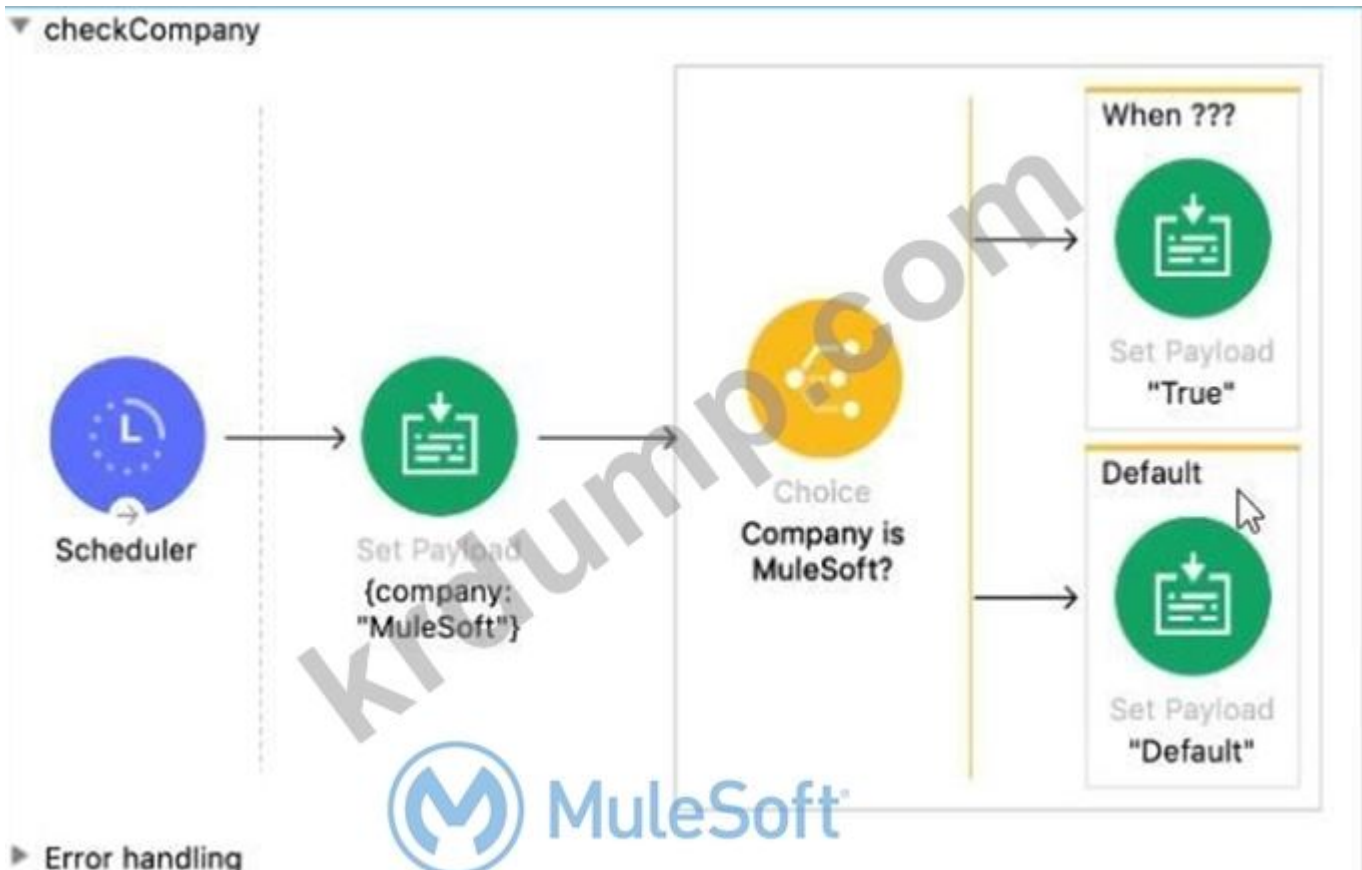
{ "oid": "1000", "itemid": "AC200", "qty": "4" } POST Mule
 FILE:CONNECTIVITY
 ORDER:NOT_CREATED?

- A. "FILE:CONNECTIVITY"
- B. "FILE:CONNECTIVITY"
- C. "ORDER:NOT_CREATED"
- D. "OTHER ERROR"

Answer: B (LEAVE A REPLY)

NEW QUESTION: 45

QUESTION: 45



▶ Error handling

```

<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>

```

Choice <when>

Mule <input type="checkbox"/>

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

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 46

200.1234

- A. ".0#" 200.1234
- B. 200.1234 {: ".0#"}
- C. ".0#" 200.1234
- D. 200.1234 {: ".0#"}

Answer: B (LEAVE A REPLY)

String {format: ".0#"} 200.1234

MCD-Level-1 DumpTop MCD-Level-1 ! DumpTop MCD-Level-1 , DumpTop MCD-Level-1 DumpTop MCD-Level-1 . DumpTop MCD-Level-1 . <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html> (235 Q&As Dumps, 30%OFF Special Discount: KrDump)

NEW QUESTION: 47

.



```

<flow name="logPayload" >
  <scheduler doc:name="Scheduler" >
    <scheduling-strategy >
      <fixed-frequency />
    </scheduling-strategy>
  </scheduler>
  <set-payload doc:name="Set to JSON" value='#[{"accounts": {"account": {"accountName": "ABC Widgets", "type": "New Customer", "stage": "Qualification"} } }]' />
  <logger level="INFO" doc:name="typeOf(payload)" message='#[typeOf(payload)]' />
</flow>

```

JSON

Logger?

- A. ". . ."
- B. ". ."
- C. ". . ."
- D. ". ."

Answer: C (LEAVE A REPLY)

NEW QUESTION: 48

□ □□ □□□ □□ □□ □□ □□□□. □□ □□□□□□□□ □□□□ □□□ □□□□□ □
 □ □□ API□ □□□ □ 2□□□ □□□□□. □□□ □ □□ □□ □□ □□ □□ □□ □□ □□
 □□□ API□ □□ □□□□□ □□□ □□ □□□□□.

MuleSoft□ □□□ □□ □□ □□□ □ □□ □□ □□□ 2□□ □□□ □ □□□□□□?

- A. □□ □□
- B. □□□ □□
- C. MuleSoft □□ □□
- D. □□ API □□□□□

Answer: B (LEAVE A REPLY)

□□:

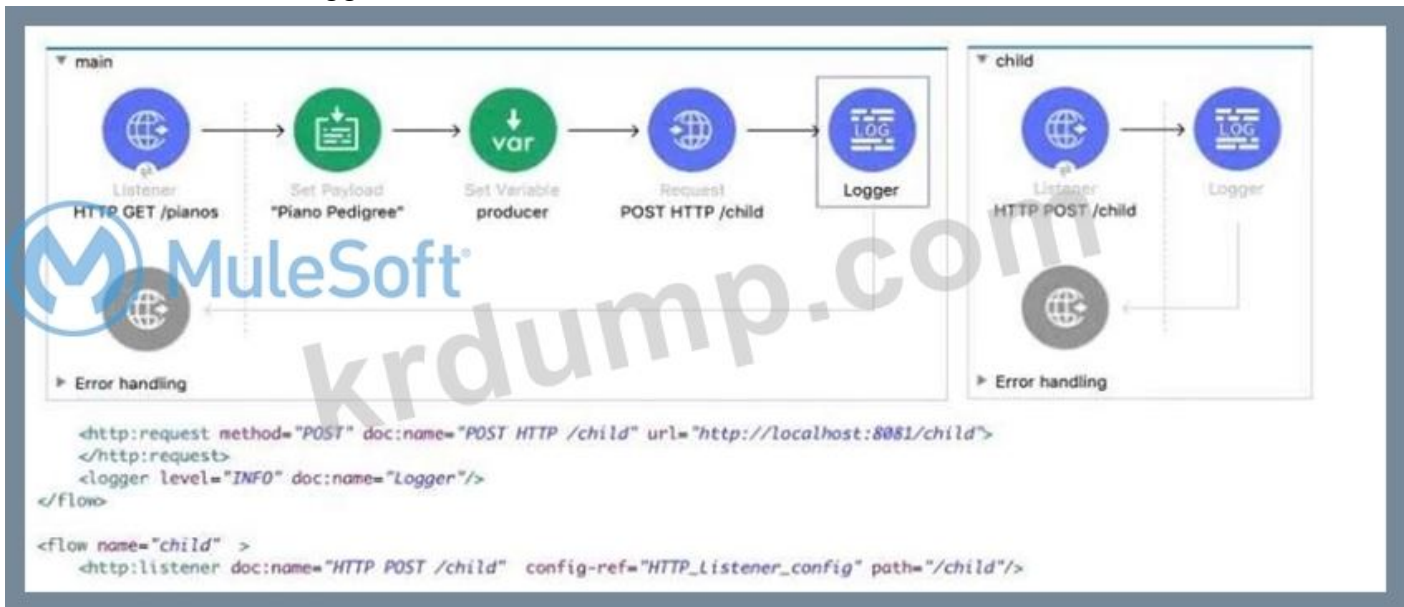
Center for Enablement□ □□□□□. □□□□□ □□ IT, LOB(□□ □□) □□, □□□ □□ □□
 □□□□□ □□□ □□□ □□□ □□□ □□□ □□□ □□ □□□ □□□, □□ □ □□□□ □
 □□ □□□□□. □ □□ □ □□ □□ API□ □□□□ □□ Center for Enablement□ □□ □□
 □□ □□□□. □□□ □□□□ □□ □ □□□□□.

NEW QUESTION: 49

□□□□ □□□□□. □□ □□□ □□ □□□ HTTP □□□ □□□□□. HTTP □□□ □ HTTP
 □□□ □□ □□□ □□□□□.

□ □□□□□□ □□□ □□□□ □□ □□ □□□□□ □□□ □□□ □□ □□□□ HTTP □□
 □□ □□□□□.

□□ □□□ □□□ Logger □□ □□□ □□□□ □ □□ □□ □□□□□□?



- A. □□□□
- B. □□□□
- □□ □□□□□
- C. □□□□
- □□ □□□□ □□□ var
- D. □□□□
- var

Answer: (SHOW ANSWER)

NEW QUESTION: 50

Anyoint Studio Mule 8082 8080 7777 6666?

- A. 8082
- B. 8080
- C. 7777
- D. 6666

Answer: D (LEAVE A REPLY)

localhost 6666 TCP port 8080.

MuleSoft link: https://docs.mulesoft.com/studio/7.5/visual-debugger-concept

NEW QUESTION: 51

http://localhost:8081



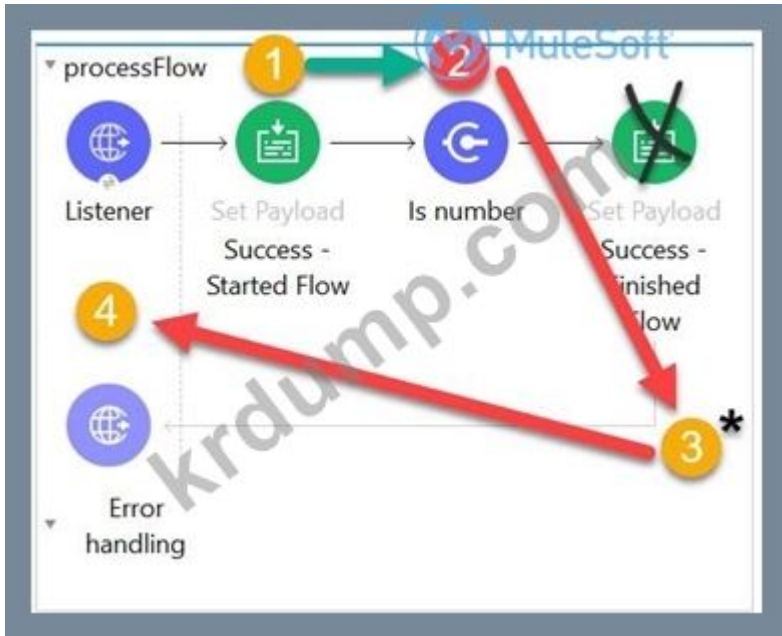
http://localhost:8081

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

Answer: (SHOW ANSWER)

1) "Before" payload before validation.

- 2) `#[error.description] = "Error Object"`
- 3) `#[error.description] = "Mule Error Object"`
- 4) `#[error.description] = "HTTP Error: 500 Internal Server Error"`



NEW QUESTION: 52

□□□ □□□□□.

```

#%RAML 1.0
title: Accounts API

/accounts:
  get:
    queryParameters:
      account_type:
        required: true
        enum:
          - "retail"
          - "commercial"
      industry:
        required: true
        enum:
          - "finance"
          - "construction"
          - "government"
  
```

□ RAML □□□□ account_type □ □□□□ □□□ □□□□ □□□□ □□□ □□□□ □□□ □.

□□ □□ □□ □□□□ □□□□ □□□ □□□□□□?

- A. /accounts/retail/finance
- B. /accounts?account_type=retail&industry=finance
- C. /accounts?account_type:retail&industry:finance
- D. /accounts/account_type=retail/industry=finance

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 53

□□□ □□□□□.

The screenshot displays the MuleSoft Anypoint Studio interface. On the left, a flow diagram shows a 'main' flow starting with a 'Listener' (HTTP: GET /), followed by a 'Request' (HTTP: GET acme.com/virgin), and then a 'Set Payload' step with the value 'Success - main flow'. Below this, an 'Error handling' section shows an 'On Error Propagate' step for 'type: HTTP:NOT_FOUND'. On the right, a detailed view of the error handling configuration shows three 'On Error Continue' steps: 1) 'type: HTTP:NOT_FOUND' leading to a 'Set Payload' step with the value 'HTTP: NOT FOUND'; 2) 'type: APP:API_RESOURCE_NOT_FOUND' leading to a 'Set Payload' step with the value 'APP: API RESOURCE NOT FOUND'; and 3) 'when: #[true]' leading to a 'Set Payload' step with the value 'Other error'. At the bottom, the XML configuration for the flow is shown:

```
<http:listener-config name="HTTP_Listener_Config" doc:name="HTTP Listener config">
  <http:listener-connection host="0.0.0.0" port="8081" />
</http:listener-config>

<flow name="main">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <http:request method="GET" doc:name="HTTP: GET acme.com/virgin" url="http://acme.com/virgin" >
    <error-mapping sourceType="HTTP:NOT_FOUND" targetType="APP:API_RESOURCE_NOT_FOUND" />
  </http:request>
```

□□ □□□ □□ □□□□ □□□□□. □ □□□□□□□ HTTP □□□□ □□□ □□□□ HTTP □□□□ HTTP:NOT_FOUND □□□ □□□□□.

□□ □□ □□□□ □□□□□?"

□□ □□ □□□□ □□□□□?

- A. HTTP: □□ □ □□

- B. □□ - □□ □□
- C. □: API □□□□ □□ □ □□
- D. □□ □□

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 54

Utility.dwl □ src/main/resources/modules □ Mule □□□□□ □□□□. Utility.dwl □□□ □□□ □ □□□□□ encryptString□□□ □□□ □□□□□. □□ □□□ □□ □□□□ encryptString □□□ □□□□ □□□ DataWeave □ □□□□□?

- A. 1. %dw 2.0
- 2. □□□□□□/json □□
- 3. □□ □□□□::□□□□
- 4. ---
- 5. □□□□::encryptString("John Smith")

- B. 1. %dw 2.0
- 2. □□□□□□/json □□
- 3. □□ □□□□::□□□□
- 4. ---
- 5. encryptString("John Smith")

- C. 1. %dw 2.0
- 2. □□□□□□/json □□
- 3. □□ □□□□.□□□□
- 4. ---
- 5. encryptString("John Smith")

- D. 1. %dw 2.0
- 2. □□□□□□/json □□
- 3. □□ □□□□.□□□□
- 4. ---
- 5. Utility.encryptString("John Smith")

Answer: B ([LEAVE A REPLY](#))

□□□
 %dw 2.0
 □□ □□□□□□/json
 □□ □□□□::□□□□

 □□□□::encryptString("John Smith")
 DataWeave 2.0 □□□ □□□ □□□□□ □□□□. □□□□ □□ DataWeave 2.0 □ Mule 4 □ □□□□. Mule 3 □□ □□ Mule 3.9 □□□□□ DataWeave □□□□ □□□□□□. □□ Mule □□□ □□ Mule □□□ □□□ □□ □□ □□□□ □□□ □ □□□□.

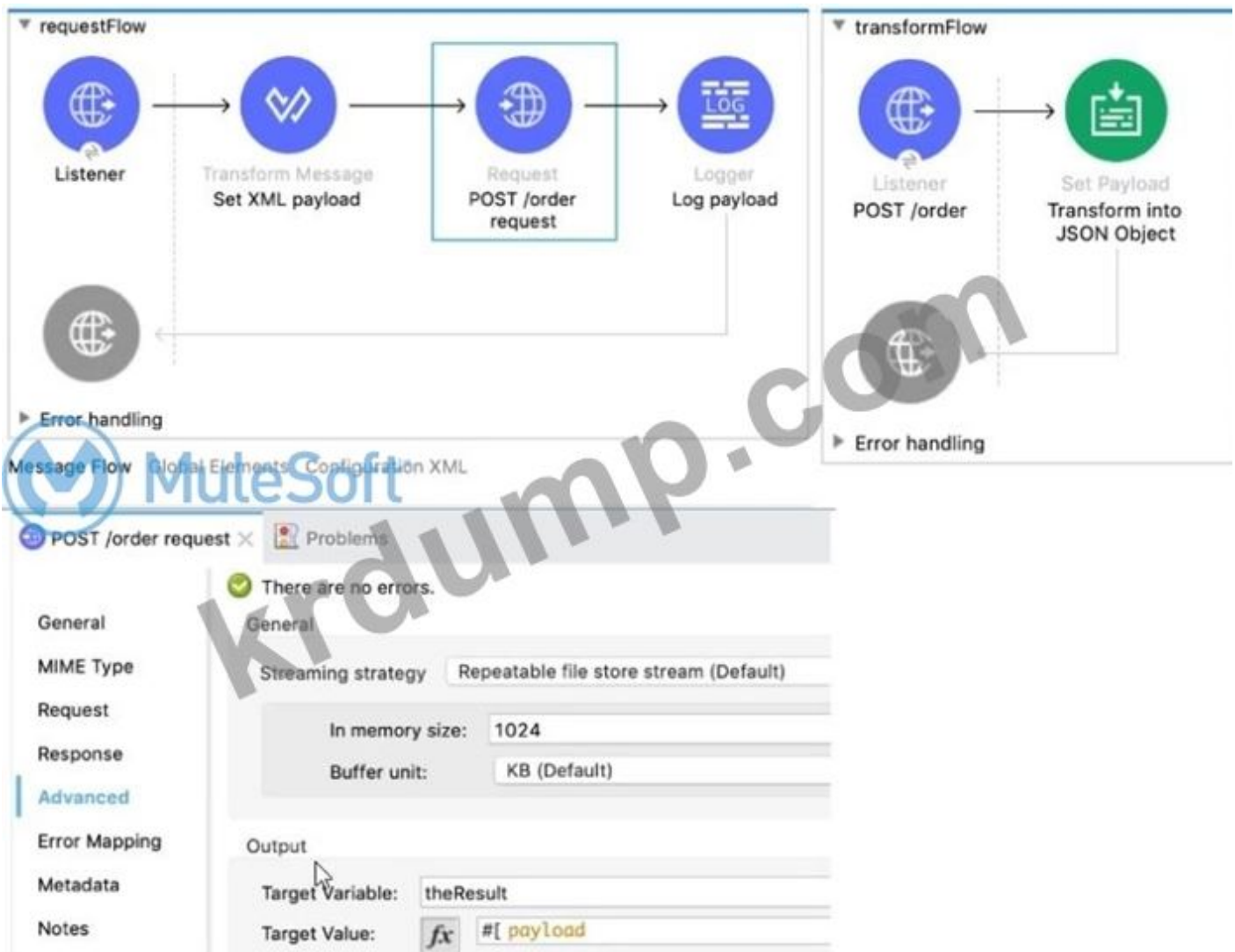
```

Core(dw::Core) □□□ □□□ DataWeave □□□□□ □□□□ □□□□□. □□ □□□ □□□
□□ DataWeave □□□□□ □□□ import □□□□ □□□□ □□□□□ □□□ □□□□ □□
□. □□ □□ □□□ □□□□.
dw::core::□□□ □□□□
□□□□ camelize, dw::core::Strings□□ □□□□ □□
dw::core::Strings□□ * □□□□
□□□ □□□□ □□□ DataWeave □□□□□□ □□ □□□ □□□□ □ □□□ □□□ □□
□ □□□. □□□□ □□□ □□ □□□ □□□□ □□□ * from□ □□□□ □□ □□□□ □□
□□□ □□□□ □□ □□□□□□ □□□ □□□ □ □□□ □□□□ □□□. □□ □□, □ □□
□□ □□□□ String □□□□ □□□ □□□ □□□□ □□□□ Strings::pluralize("box")□ □□
pluralize □□□ □□□□□.
□□
%dw 2.0
dw::core::□□□ □□□□
□□ □□□□□□/json
---
{ '□□□': □□□::pluralize("□□") }

```

NEW QUESTION: 55

□□□□ □□□□□.



requestFlow HTTP XML HTTP .
 HTTP .
 JSON HTTP .
 Result .
 HTTP ?

- A. Java
- B. JSON
- C.
- D. XML

Answer: D (LEAVE A REPLY)

NEW QUESTION: 56



```

<flow name="main">
  <http:listener doc:name="HTTP: POST /" config-ref="HTTP_listener_config" path="/" />
  <try doc:name="Try" >
    <validation:is-null doc:name="payload" value="#[payload]" message="Validation Error"/>
    <error-handler >
      <on-error-propagate enableNotifications="true" logException="true" doc:name="On Error Propagate">
        <set-payload value="Error - Try scope" doc:name="Error - Try Scope"/>
      </on-error-propagate>
    </error-handler>
  </try>
  <set-payload value="Success - main flow" doc:name="Success - main flow" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - main flow" doc:name="Error - main flow" />
    </on-error-continue>
  </error-handler>
</flow>

```

Try scope is used to handle errors in a flow. It is used to catch errors from a processor and handle them in a specific way. In this example, the Try scope is used to catch errors from the validation:is-null processor and set the payload to "Error - Try scope". The On Error Propagate error handler is used to catch errors from the Try scope and set the payload to "Error - Try scope". The On Error Continue error handler is used to catch errors from the Try scope and set the payload to "Error - main flow".

- A. 00 - 00 00
- B. 00 - 00 00
- C. 00 - 00 00
- D. 000 00 00

Answer: A (LEAVE A REPLY)

00 0000 On Error Continue 0 0000 00 0000 0000. 0000 00 0000 0000
 0000 0 On Error Continue 0000 0000 00 0000 00 0 0000 0000 0000
 00 0000 0000. 0000 0000 000000 0000 000000 00 - 00 0000 00
 000.

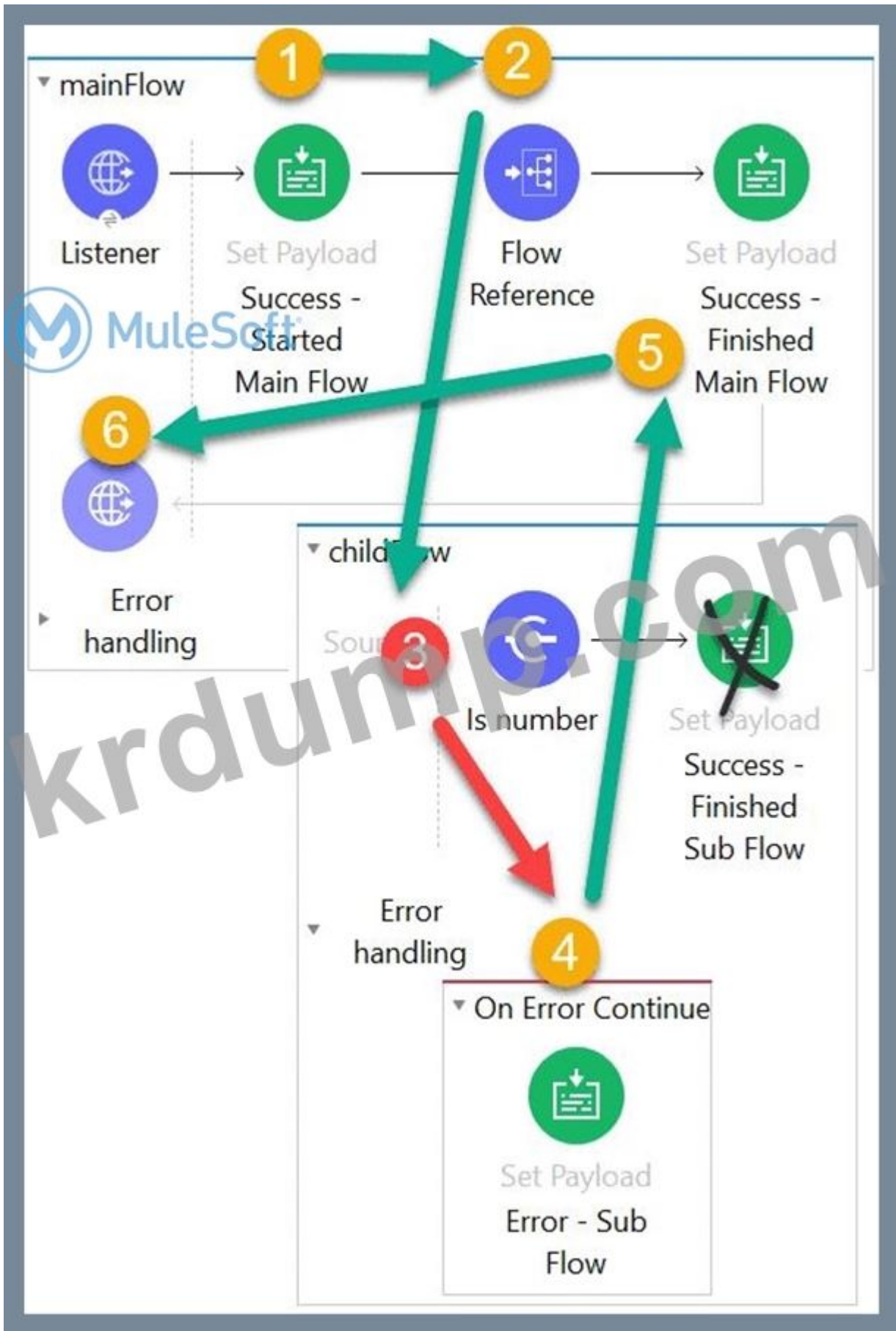
0000 0000 00 - 00 000000.

- 1) HTTP 0000 00 00
- 2) 00 0000 00 0000 000000.
- 3) Is Number 0000 000000 000000 0000 0000 00 0000 000000. 00 0
 0 00 00

#[error.description] = "000000 0000 INTEGER 00 0000"

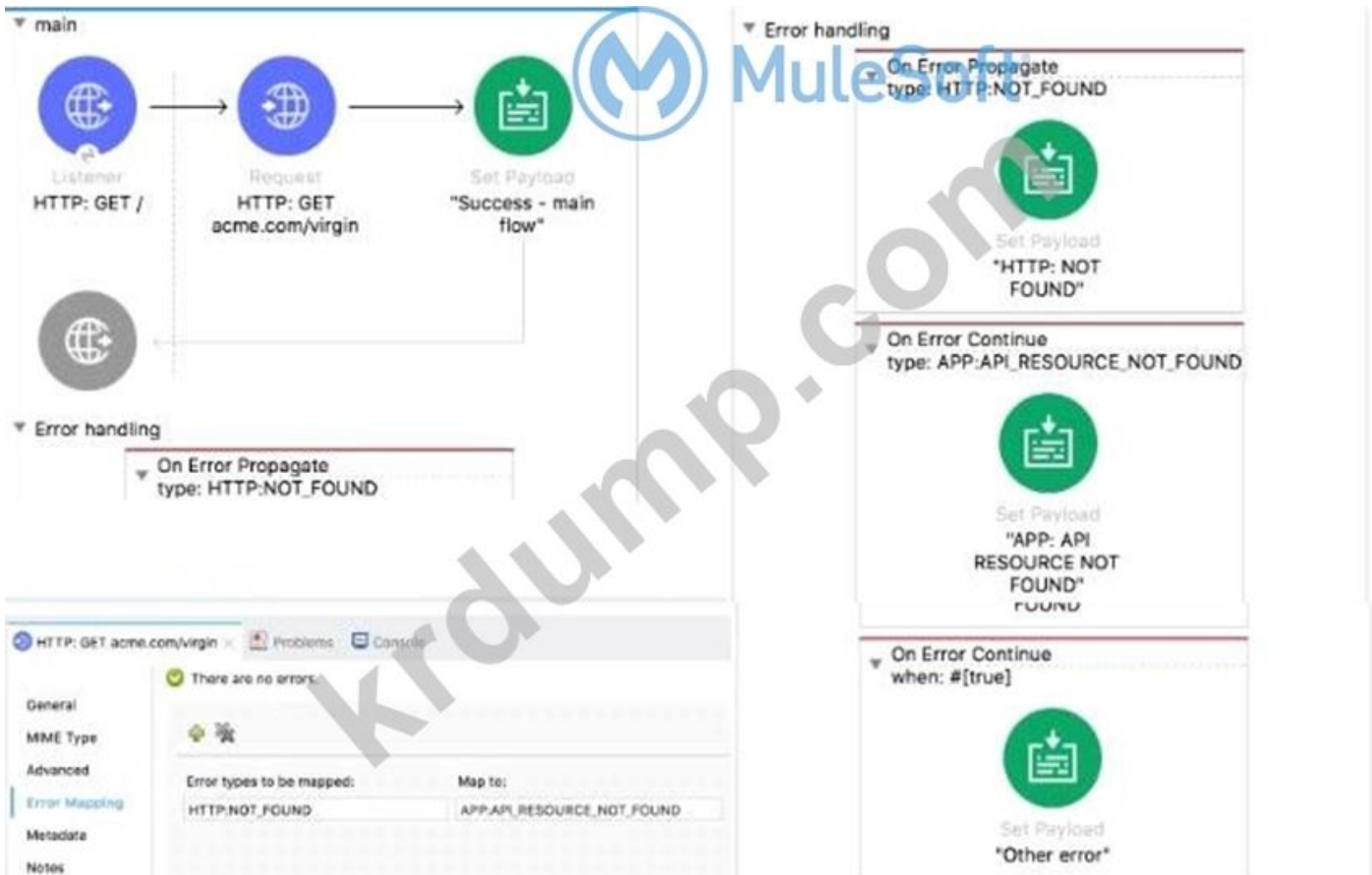
#[error.errorType] = VALIDATION:INVALID_NUMBER

- 4) On Error Continue 0 0000 000000. 000000 "Error - Sub Flow" 0 000000.
- 5) "Error - Sub Flow" 0 00 0000 0000 0000 00 0000 000000. 000000 0000 00
 0000. 000000 "Success - Finished Main Flow" 0 00000000.
- 6) "Success - Main Flow" 0 HTTP 00 000000 000000 000000. HTTP 00 00: 200
 00 0000 0 0 0000 Mule Message 00 000000 0000 0 00 0000 (RED in, GREEN
 out) 0 On Error Continue 0000 00 0000 000000 0000 00 000000 on error continue
 0 200 00 000000 000000 0000 0000 000000 0000. mainFlow 0 000000 childFlow
 0 0000 0000 0000 0000 00 00 0000 mainFlow 0 0000 0000000000.



NEW QUESTION: 57

□□□ □□□□□.



□□ □□ □□ □□□□ □□□□□. □ □□□□□□ HTTP □□□□ □□□ □□□□ HTTP □□□□ HTTP:NOT_FOUND □□□ □□□□□.

□□ □□ □□□□ □□□□□?"

□□ □□ □□□□ □□□□□?

A. □: API □□□□ □□ □ □□

B. HTTP: □□ □ □□

C. □□ □□

D. □□ - □□ □□

Answer: A (LEAVE A REPLY)

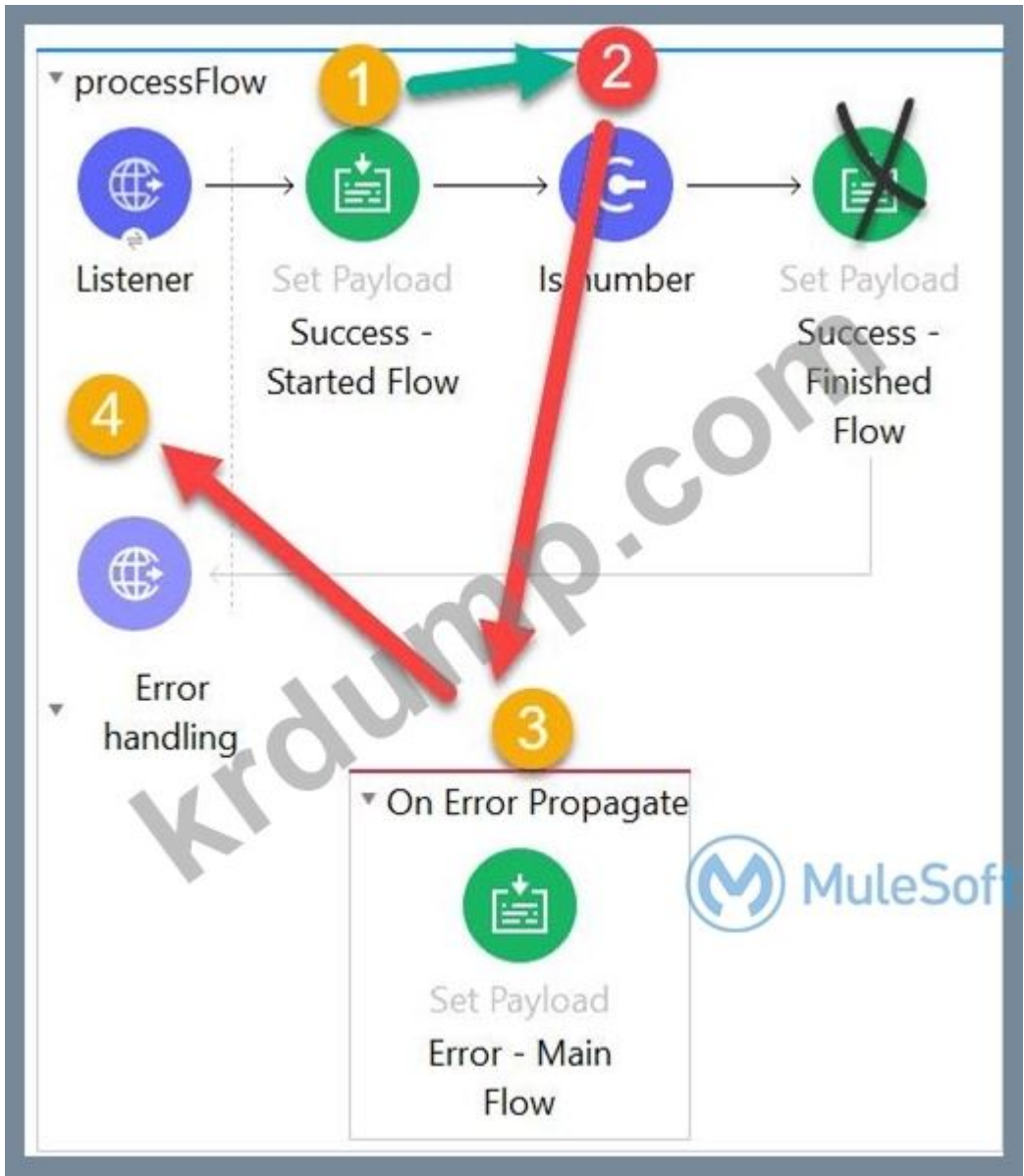
□□□ □: API □□□□ □□ □ □□□□.

1) □ □□□□□□ HTTP Listener□ □□□ □□□□□.

2) HTTP □□□□ "HTTP:NOT_FOUND" □□□ □□□□ □□□ □□□□□.

3) □□ □□ □ □□ □□ □□□□ □□□ □□□□□. □ □□ HTTP:NOT_FOUND □□□ □□ □ □□ □□ APP:API_RESOURCE_NOT_FOUND□ □□□□□. □ □□ □□□□□ □□□□ □ APP:API_RESOURCE_NOT_FOUND□ □□□□□.

4) "APP:API_RESOURCE_NOT_FOUND." HTTP 500 HTTP



NEW QUESTION: 58

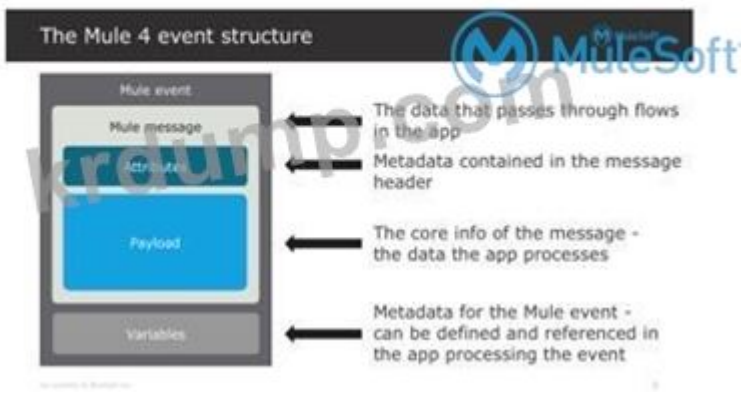
□□□ □□□□□.



Mule ?

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

Answer: A (LEAVE A REPLY)



NEW QUESTION: 59
 .



```

<flow name="main">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <http:request method="POST" doc:name="HTTP: POST /data" url="http://localhost:8081/data"/>
  <jms:publish-consume doc:name="JMS: num1" config-ref="JMS_Config" destination="num1"/>
  <jms:publish doc:name="JMS: num2" config-ref="JMS_Config" destination="num2"/>
  <set-payload value="#[payload + 1]" doc:name="payload + 1" />
</flow>
  
```

http://localhost.8081/□ □□ □□□□ □□□□ □□□□ □□□□

http://localhost;8081/□ □□ □□□□ □□ □□□□□ □□□□□□?

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 60

HTTP Listener □□□ APIkit □□□□ □□ □□□ □□□□□□ □□ □□ □□□ □□□□□□?

- A. /(*)
- B. /
- C. /()
- D. /'

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 61

□□ □ HTTP Listener□ □□ □□□ □□ □□ □□□□□?

- A. □□
- B. □□□□ □□
- C. □□□ □□□ HTTP □□
- D. □□□ □□□ HTTP □□□

Answer: B (LEAVE A REPLY)

□□□□ □□□ □□□ □□□□□. □□ □□ □□□□ □□□ □□ HTTP □□□□ □□□□□. □□□□ □□ □□□□□.

MCD-Level-1 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ MCD-Level-1 □ □! DumpTop □ □□ **MCD-Level-1** □□ □□□ □□□□□□, DumpTop MCD-Level-1 □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop MCD-Level-1 □□□ □□□□□. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html> (235 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 62

Mulesoft□ □□ API □□ □□ □□□ □□ □□□ □□□□□□ □□□□□ □□□□□□. □□ □□□ □□□□□□ □□□□□□ □□ API□ □□□□ □□□□ □□, □□ □ □□□□ □□ API□ □□□□ □□□. □ API□ □□ □□ API□ □□□□ □□□ □ □□ □□ □□□ □□□□ □□□.

Mulesoft□ □□ API □□ □□ □□□ □□□ □ □□□ API□ □□ □□□ □□□ □□□□□?

- A. □□□□ □□□
- B. □□□ □□□
- C. □□□
- D. □□□ □□□

Answer: (SHOW ANSWER)

□□□ □□ □□□□□□□ □ □□ □□□ API □□ □□□ □□ Mulesoft□ □□ □□ □□□ □□ □□□□ □□□ □□□ □□□ □□□ □□□□□.

NEW QUESTION: 63

API □□□□ □□ API □□ □□□ API □□□□□ Anypoint Exchange□ API □□□ □□□□ API □□□□□ □□□□□. API □□□□□ CloudHub□□ □□ □ □□□□ API □□□□□ □ □□□□.

SLA □□ □□□ □ API □□□□□ □□ API □□□□□ □□□□□□. □□ API □□□□□ API □□□□□ □□□□□ □□□ □□□ □□ □□□ □□□□□ ID□ □□□ □□ □□□ □□□□ □□ □ □□□□?

- A. Anypoint Exchange□ □□ □□ API □□□□□ API □□□□□ □□ □□□ □□□□□ □□□□ □□□□

- B. Anypoint Studio API APIkit API
- C. Anypoint Studio Rest Connect API
- D. API API API API

Answer: A (LEAVE A REPLY)

- * Anypoint Platform API ID
- * API API API API API API
- * ID API API API API API

----- API: API API Anypoint Exchange API API
 API API API API

NEW QUESTION: 64

- API FTP API API API API API API API API API?
- A. API API API API API
- B. API API API
- C. API API API API API
- D. API API API API API

Answer: A (LEAVE A REPLY)

NEW QUESTION: 65

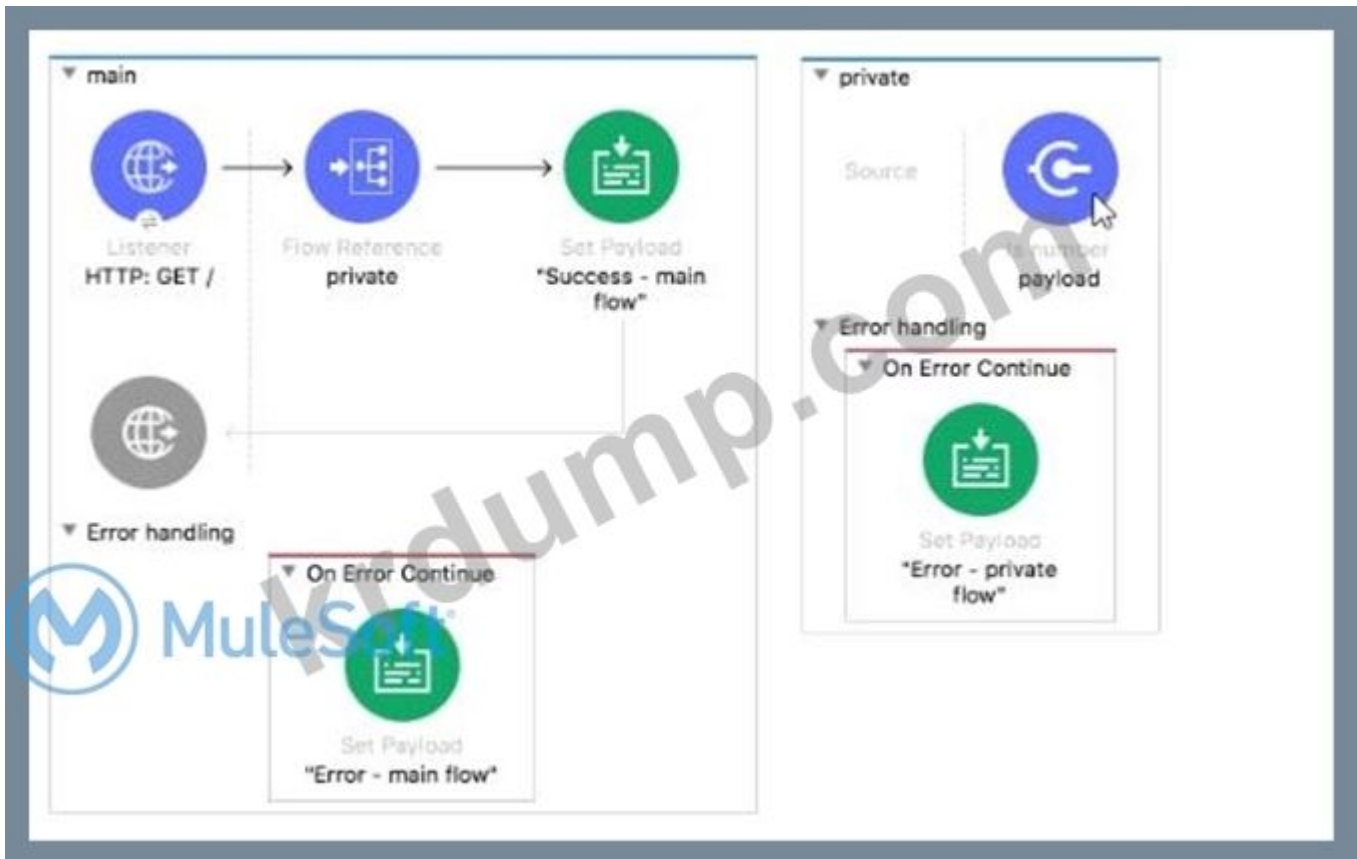
- DataWeave API API API API API
- A. API API
- B. API API
- C. API
- D. API API API

Answer: B (LEAVE A REPLY)

API API
 DataWeave Mule API API API API API API API API MuleSoft API API
 API. DataWeave Mule API API API API API Mule API API API
 API API.
 API API: <https://docs.mulesoft.com/mule-runtime/4.3/dataweave>

NEW QUESTION: 66

- API API API



□□ □□□ □□□ □□ □□ □□□□ □□□ □□□□□. □□ □□□□ HTTP □□□□ □□ □□□□□ □□□ □□ □□ □□ □□□□ □□□□□?

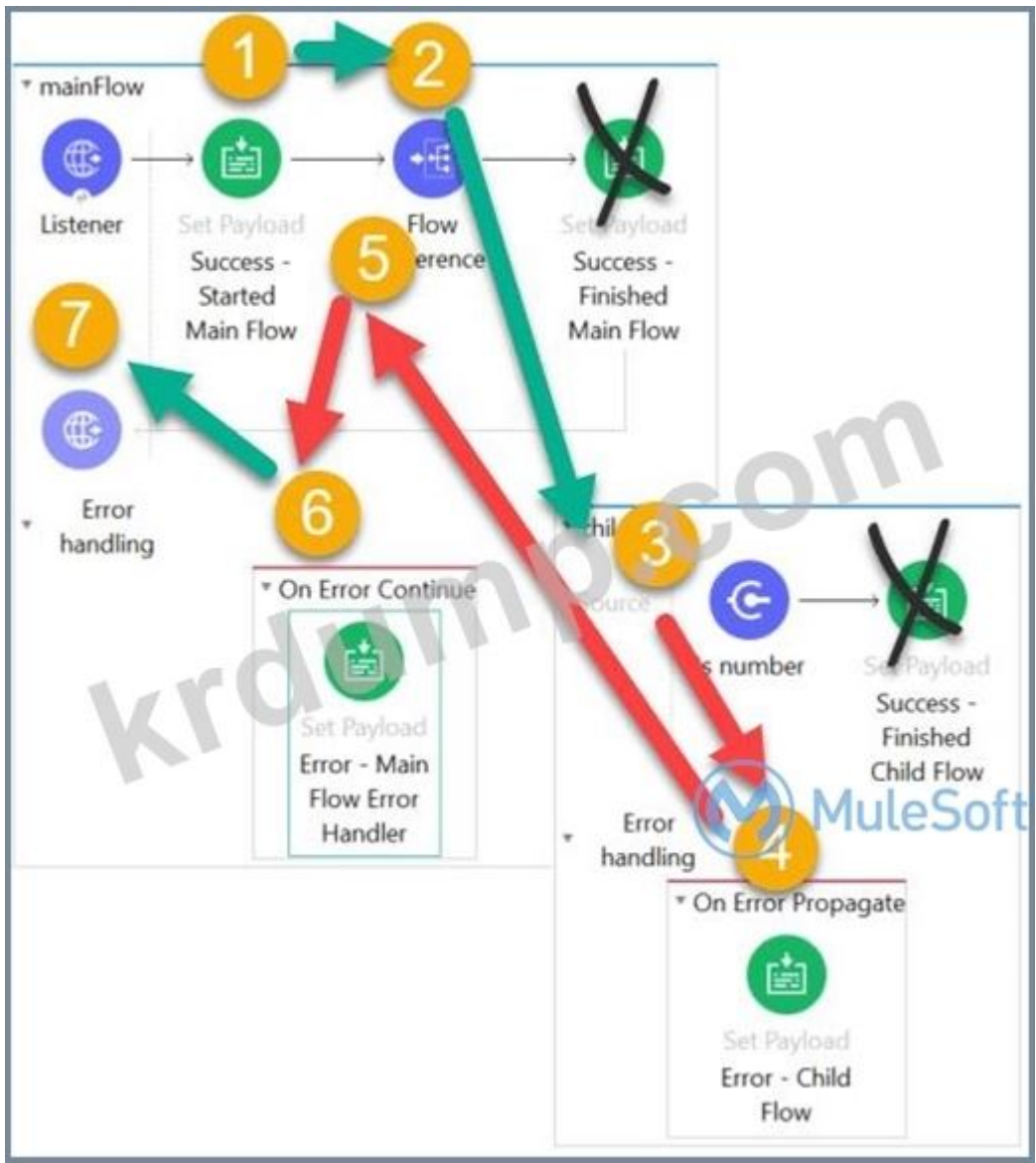
- A. □□ - □□ □□
- B. □□ - □□ □□
- C. □□ - □□ □□
- D. □□□ □□ □□

Answer: B (LEAVE A REPLY)

□□□ □□ □□ □□□ □□□ □□□□ □□ On Error Propagate □□□ □□ □□□ □□ □□ □□□ "Error-main flow" □ □□□□ □□ □□ □□ □□□□ □□ □□□□ □□□ □□ □□□ □□. □□□ □□□ □□ - □□ □□□□□.

- 1) HTTP □□□□ □□□ □□□
- 2) □□ □□□ □□□□□.
- 3) □□□□□ null □ □□□ □□□ Try □□□ □□□ □□□ □□ □□□ □□ □□□ □□□□ □.
- 4) On Error Propagate □ □□□ □□□□□. □□□□□ "□□ - □□ □□" □ □□□□□.
- 6) "□□ - □□ □□" □ '□□ □ □□' □□□□ □□□□□. □□ □□ □□□ □□□□□. □□□ □□ "□□ - □□ □□" □□ □□□
- 7) "□□ - □□ □□" □ HTTP □□ □□□□ □□□□□ □□□□□. HTTP □□ □□: 200

 ----- Reference Diagram □□□ □□□ □□□ □□□ □□□□□. □□ □□ □□□□ □□ □□□ □□ □□□ □□□ □□□ □ □□□ □ □□□□.



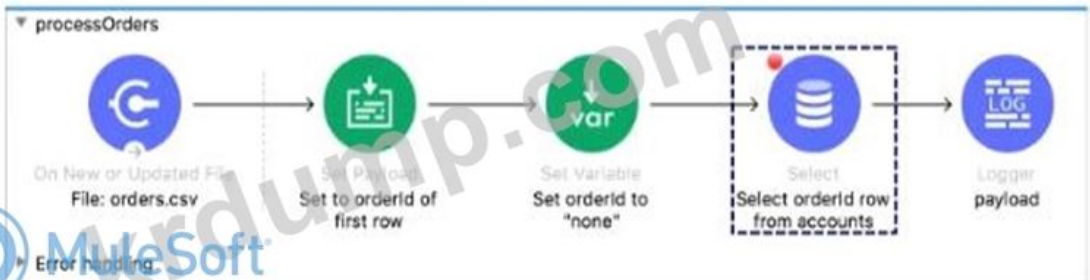
NEW QUESTION: 67

□□□□ □□□□□.

orders.csv

```

orderId,account
100, partnerA
101, acme.com
102, mybank.com
103, onlineSales
  
```



orders.csv □□□ □□ □□ □□□□□□□□ □□□ □□□□□ □□□□□. Mule □□□□□

□□ Any point Studio□□ □□□□□ □□□□□ □□□□□.

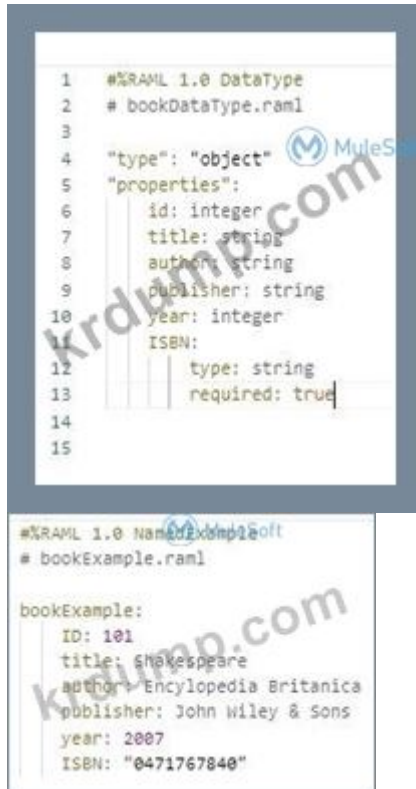
□ □□□□□ □□□□ □□□□ □□□□□ □□□□□?

- A. 100
- B. □□□□□□ □□
- C. "□□"
- D. □□ CSV □□

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 68

□□□ API□□ □□□ □ □□□ □□□ □ □□□□ □ □□ RAML □□□ □□□□□□.
□□□ □□□ □□□□ □ □□□ RAML□ □□□□□?



- A.** 1. #%RAML 1.0
2. □□: □
3. □□:
4. □: ABC/Examples/bookDataType.raml
5. /□:
6. □□□:
7. □□:
8. □□□□□□/json:
9. □□: □
10. □:
11. □□: ABC/Examples/bookExample.raml
12. □□:
201□ 13□:
14. □□:
15. □□□□□□/json:
16. □:
17. □□□: □ □□□
- B.** 1. #%RAML 1.0
2. □□: □

3. : !bookDataType.raml
4. /:
5. :
6. :
7. /json:
8. :
9. :
10. : !bookExample.raml
11. :
- 201 12:
13. :
14. /json:
15. :
16. :

C. 1. #%RAML 1.0

2. :
3. : bookDataType.raml
4. /:
5. :
6. :
7. /json:
8. :
9. :
10. : bookExample.raml
11. :
- 201 12:
13. :
14. /json:
15. :
16. :

D. 1. #%RAML 1.0

2. :
3. : bookDataType.raml
4. /:
5. :
6. :
7. /json:
8. :
9. :
10. : bookExample.raml

- 11. □□:
- 201□ 12□:
- 13. □□:
- 14. □□□□□□/json:
- 15. □:
- 16. □□□: □ □□□

Answer: (SHOW ANSWER)

```
* RAML □□□□ "API □□□ □□"□□ □□□ □ □□ □□ □□□□ □□□□. □□□
"□□□□ □□□" □□.
□□□ □□□□□ □□□ □□ RAML □□□ □□□ □□□□.
* !includes□ □□ RAML□ □□□□ □□ □□ API □□□ □□□ □ □□□□. □□ □□ □□□
□ □□□□ □ □□□ □□ □□□ □□□□ □□□□□□.
* □□□ □□□ RAML □□□ □□ □□ □□□ □□ !include□ □□□□ □□ RAML □□□□□
□□□ □ □□□□.
□□:
□: !bookDataType.raml □□ □
□:
□□: !bookExample.raml □□
* □□□□□
□□: D
```

NEW QUESTION: 69

□□ □□□ RAML□ URL □□□□□ □□ orderId□ □□ □□ □□□ □□ □□ □□□ □□□ □□□□ □□□?

A)

```
/orders:
  /{orderId}:
    get:
```

B)

```
/orders:
  /orderId:
    get:
```

C)

```
/orders:
  get:
    /{orderId}:
```

D)

```
/orders:
  get:
    /orderId:
```

A. □□ A

- B. □□ D
- C. □□ C
- D. □□ B

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 70

□□□ □□□□□.

```

<http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener Config">
  <http:listener-connection host="0.0.0.0" port="8081" />
</http:listener-config>

<flow name="main">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <http:request method="GET" doc:name="HTTP: GET acme.com/virgin" url="http://acme.com/virgin">
    <error-mapping sourceType="HTTP:NOT_FOUND" targetType="APP:API_RESOURCE_NOT_FOUND" />
  </http:request>

```

□□ □□□ □□ □□□□ □□□□□. □ □□□□□□□ HTTP □□□□ □□□ □□□□ HTTP □□□□ HTTP:NOT_FOUND □□□ □□□□□.

□□ □□ □□□□ □□□□□?"
 □□ □□ □□□□ □□□□□?

- A. HTTP: □□ □ □□
- B. □: API □□□□ □□ □ □□
- C. □□ - □□ □□
- D. □□ □□

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 71

Mule ActiveMQ JMS . Mule Anypoint Studio Anypoint Studio . Mule Anypoint Studio . Anypoint Studio JAR ?

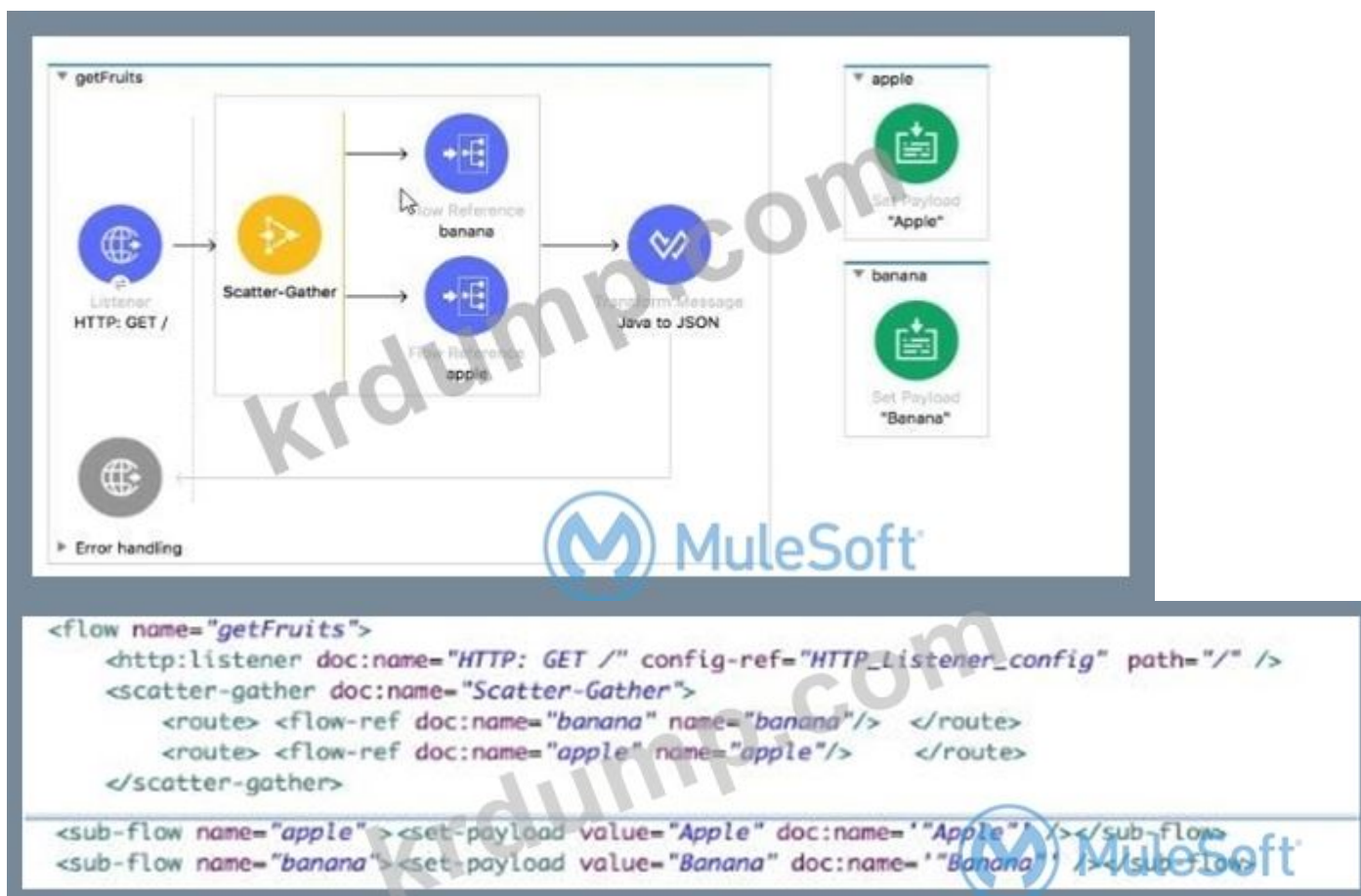
- A. Attach project sources
 Include project modules and dependencies
- B. Attach project sources
 Include project modules and dependencies
 Attach project sources
 Include project modules and dependencies
- D. Attach project sources
 Include project modules and dependencies

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 72

.



http://localhost:8081/ HTTP GET /.
 ?

A) `['Banana', 'Apple']`

B) `{ "0": "Banana", "1": "Apple" }`

C) `{ "attributes": ..., "payload": ['Banana','Apple'] }`

D)

```

{
  "0": {
    "attributes": ...,
    "payload": "Banana"
  }
  "1": {
    "attributes": ...,
    "payload": "Apple"
  }
}

```

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

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 73

payload

```

"transaction_id": "SS-4848-44KK-4SYQ",
"account_id": "KA-382-SKD44",
"name": "Max Mule",
"position": "sell"

```



```

<flow name="writeRecords" >
  <http:listener doc:name="HTTP: POST /records" config-ref="HTTP_Listener_config"
    path="/records" allowedMethods="POST"/>
  <ee:transform doc:name="Add write_date">
    <ee:message >
      <ee:set-payload ><![CDATA[%dw 2.0
        output application/json
        ---
        payload ++ {"write_date": now()}]]>
      </ee:set-payload>
    </ee:message>
  </ee:transform>
  <file:write doc:name="File: records.csv" path="file-store/records.csv">
    <file:content ><![CDATA[#payload]]></file:content>
  </file:write>
</flow>

```

records.csv ?

- A. JSON
- B. CVS
- C.
- D.

NEW QUESTION: 77

□□□□ □□□□□.

The image displays the MuleSoft interface. On the left, a flow diagram for 'main' shows a sequence of steps: 'Listener HTTP: GET /', 'Set Payload "Success - Begin"', 'Request HTTP: GET /data', and 'Set Payload "Success - End"'. Below this, an 'Error handling' section shows an 'On Error Continue' block with a 'Set Payload "Error"' step. On the right, the console window shows 'HTTP: GET /' with a status of 'There are no errors.' and response details including 'Body: #[payload]' and 'Error Response Body: #[output text/plain --- error.description]'. A large watermark 'krdump.com' is overlaid across the interface.

```
<flow name="main">  
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_listener_config" path="/" />  
  <set-payload value="Success - Begin" doc:name="Success - Begin" />  
  <http:request method="GET" doc:name="HTTP: GET /data" url="http://mu.learn.mulesoft.com/data"/>  
  <set-payload value="Success - End mainFlow" doc:name="Success - End" />  
  <error-handler >  
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue">  
      <set-payload value="Error" doc:name="Error" />  
    </on-error-continue>  
  </error-handler>  
</flow>
```

- □□□□□□ HTTP □□□□ □□□ □□□□ HTTP □□□□ □□□ □□□□□.
- □□□□□□ □□□□ □□□□ □ □□ □□□ □□□□□?
- □□□□□. □ □□□□□□ HTTP □□□□ □□□ □□□□ HTTP □□□□ □□□ □□□□□.
- □□□□□□ □□□□ □□□□ □ □□ □□□ □□□□□?
- A. □□ □□ □□: □□, □□ □□ □□ □□ □□ □□: 500
 - B. □□ □□: "□□ - □□" □□ □□ □□ □□: 200
 - C. □□ □□: "□□ - □□* □□ □□ □□ □□: 200
 - D. □□ □□: "□□" □□ □□ □□ □□: 200

Answer: D (LEAVE A REPLY)

NEW QUESTION: 78

□□□□ □□□□□.

Payload

```

{
  "transaction_id": "55-4848-44KK-45YQ",
  "account_id": "KA-382-5KD44",
  "name": "Max Mule",
  "position": "sell"
}

```

```

<flow name="writeRecords">
  <http:listener doc:name="HTTP: POST /records" config-ref="HTTP_listener_config">
    path="/records" allowedMethods="POST"/>
  <ee:transform doc:name="Add write_date">
    <ee:message >
      <ee:set-payload ><![CDATA[<w 2.0
output application/json
---
payload ++ {"write_date": now()}]]>
      </ee:set-payload>
    </ee:message>
  </ee:transform>
  <file:write doc:name="File: records.csv" path="file-store/records.csv">
    <file:content ><![CDATA[#<payload]]></file:content>
  </file:write>
</flow>

```

□□□ □□□ □ records.csv □□□ □□□ □□□□□□?

- A. JSON □□□□
- B. □□ □□□
- C. □□□□□ CVS□ □□
- D. □□□□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 79

□□ □□□ RAML□ URI □□□□□ □□ customerId□ □□ □□□ □□ □□ □□□ □□□□

□?

A. 1. /□□:

2. /get:

3. /□□ ID:

B. 1. /□□:

2. /{□□ ID}:

3. □□:

C. 1. /□□:

2. /□□ ID:

3. □□:

D. 1. /□□:

2. □□:

3. /{□□ ID}:

Answer: (SHOW ANSWER)

□□□ □□□ □□□□ □□□ □□□ □□□□.

/□□:

/{ID}:
 :
 :

NEW QUESTION: 80

Dataweave Map : ?

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

Answer: D (LEAVE A REPLY)

: () : : : .

MuleSoft : <https://docs.mulesoft.com/mule-runtime/4.3/dataweave-cookbook-map> : : : : : Dataweave : . : : : : . : : : : : . : : : : .

Important Operators			
Operator	Can be applied on input which is/are	Output type	Used for
map	On Arrays only	Array	Array of Object
mapObject	On Objects only	Object	Output is an Object
reduce	On Arrays	Anything	To reduce into given expression
pluck	On Objects	Array	Same as mapObject, only difference is the output is returned as array instead of Object
flatten	On arrays	Single set of Array	Turns into set of subarrays to single array

NEW QUESTION: 81

: : . : : : : , : : : . : : : : : : : : : : : ?

□□□□ □□□□.



Mule □□□□□□□ http://acme.com/order/status □ http://vacme.com/customer/status URL□□
 □ □□□□□□ GET □□□ □□□□ REST API□ □□□□□.
 □ □ URL□ □ □□□□□ □□ □□□□ □□ HTTP GE~ □□□ □□□ □□ □□ □□ □□
 □ □ □□□□?

- A. */□□
- B. *□□
- C. ?[□□,□□]/□□
- D. *{□□,□□}/□□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 84

□□□□ □□□□.



```

<flow name="main" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_Listener_config" path="/" />
  <flow-ref doc:name="private" name="private"/>
  <set-payload value="Success - main flow" doc:name="Success - main flow" />
  <error-handler>
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - main flow" doc:name="Error - main flow" />
    </on-error-continue>
  </error-handler>
</flow>

<flow name="private" >
  <validation:is-number numberType="INTEGER" doc:name="payload" value="#[payload]"
  message="Validation Error" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - private flow" doc:name="Error - private flow" />
    </on-error-continue>
  </error-handler>
</flow>

```

Which of the following is the correct configuration for the main flow?

- A. Success - main flow
- B. Error - main flow
- C. Success - private flow
- D. Error - private flow

Answer: [\(SHOW ANSWER\)](#)

NEW QUESTION: 85

Which of the following is the correct configuration for the main flow?

The screenshot shows the MuleSoft interface with an XML input on the left and a JSON output on the right. The XML input is:

```

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

```

The JSON output is:

```

[
  {
    "index": 0,
    "orderId": "592",
    "itemName": "T-shirt Navy",
    "lineItemPrice": 20
  },
  {
    "index": 1,
    "orderId": "972",
    "itemName": "Cargo Shorts",
    "lineItemPrice": 60
  }
]

```

Which of the following is the correct configuration for the main flow?

- A)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.orderId,
  itemName: value.item,
  lineItemPrice: (value.price as :number) * (value.quantity as :number)
})
```

B)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.@orderId,
  itemName: value.item,
  lineItemPrice: (value.price as Number) * (value.quantity as Number)
})
```

C)

```
payload.order.*item map ( (value,index) -> {
  index: index,
  orderId: value.@orderId,
  itemName: value.item,
  lineItemPrice: (value.price as :number) * (value.quantity as :number)
})
```

D)

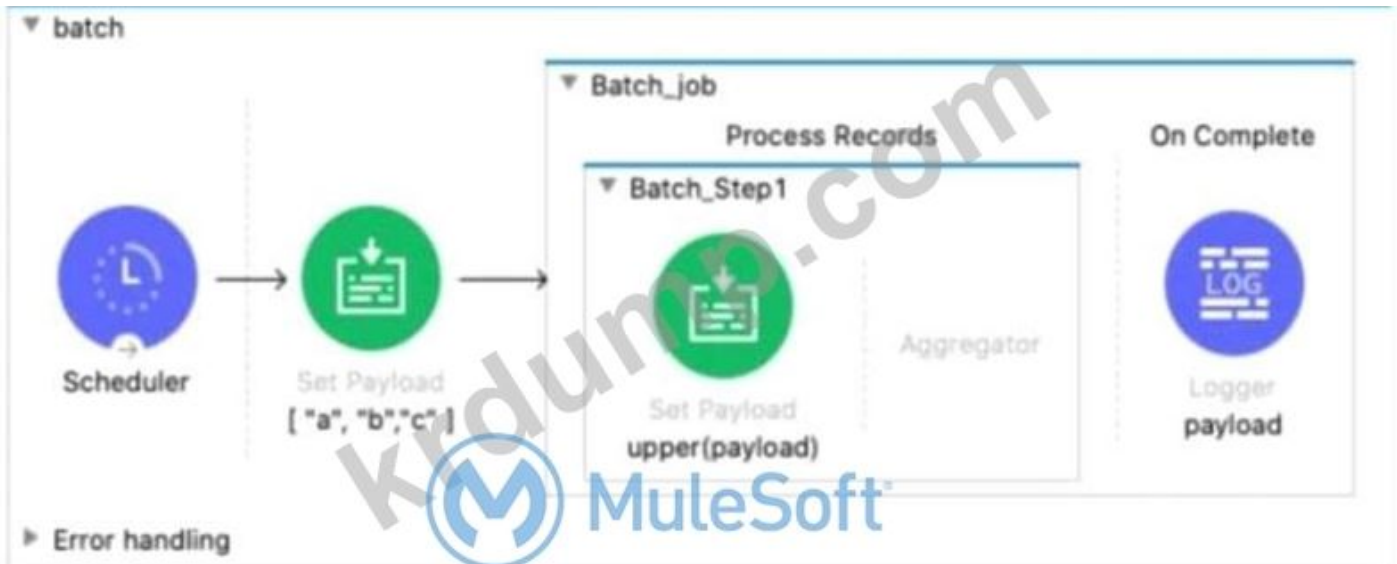
```
payload.order.*item map( (value,index) -> {
  index: index,
  orderId: value.orderId,
  itemName: value.item,
  lineItemPrice: (value.price as Number) * (value.quantity as Number)
})
```

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

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 86

.



```

<flow name="batch" >
  <scheduler doc:name="Scheduler" >
    <scheduling-strategy >
      <fixed-frequency frequency="10000" />
    </scheduling-strategy>
  </scheduler>
  <set-payload value='#[["a", "b", "c"]]' doc:name='["a", "b", "c"]' />
  <batch:job jobName="Batch_job" >
    <batch:process-records >
      <batch:step name="Batch_Step1" >
        <set-payload value='#[upper(payload)]' doc:name='upper(payload)' />
      </batch:step>
    </batch:process-records>
    <batch:on-complete >
      <logger level="INFO" doc:name="payload" message="#"[payload]" />
    </batch:on-complete>
  </batch:job>
</flow>

```

□□ □□ □□□ □□□ □□□□□□.

□□ □□ □□□ □□ □□□□ □□□ □□□ □ □□□□ □□ □□□ □□□□

□?

A)

```

Total Records processed: 1
Successful records: 1
Failed Records: 0
payload: ["A", "B", "C"]

```

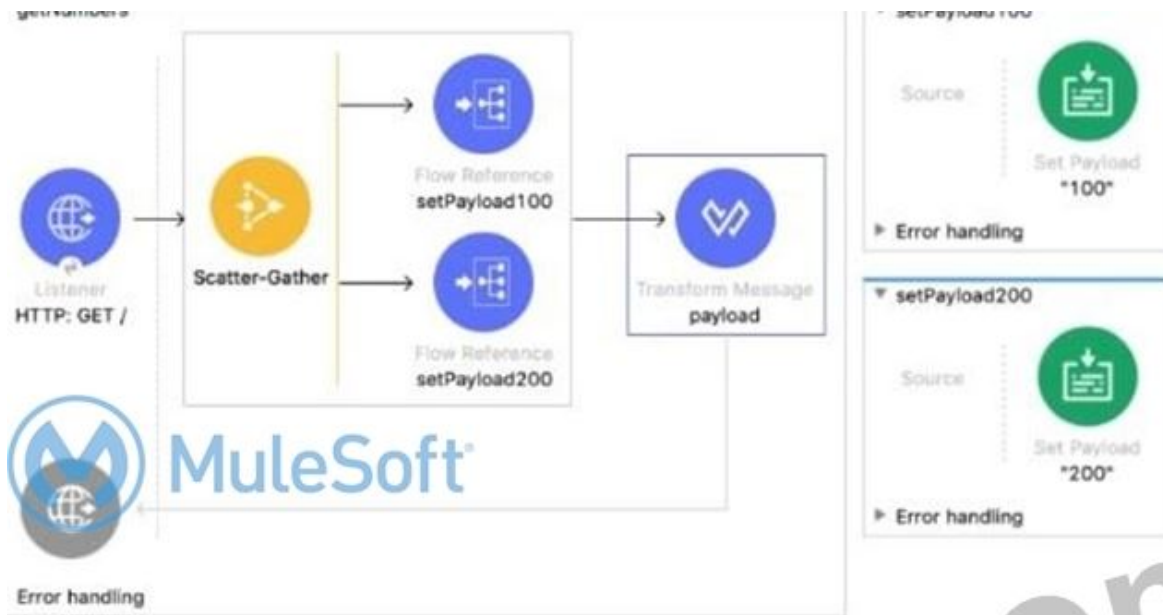
B)

```

Total Records processed: 3
Successful records: 3
Failed Records: 0
payload: ["A" "B" "C"]

```

C)



```

flow name="getNumbers" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_listener_config" path="/" />
  <scatter-gather doc:name="Scatter-Gather" >
    <route >
      <flow-ref doc:name="setPayload100" name="setPayload100"/>
    </route>
    <route >
      <flow-ref doc:name="setPayload200" name="setPayload200"/>
    </route>
  </scatter-gather>
  <ee:transform doc:name="payload">
    <ee:message >
      <ee:set-payload ><![CDATA[%dw 2.0
        output application/json
        ---
        payload]]></ee:set-payload>
    </ee:message>
  </ee:transform>
</flow>

```

```

flow name="setPayload100" > <set-payload value='#[ "100" ]' doc:name='100' /></flow>
flow name="setPayload200" > <set-payload value='#[ "200" ]' doc:name='200' /></flow>

```

Scatter-Gather□ □ □□□ □□□□□ □□□□ □□□ □□□□□. HTTP Listener□ □
 □ □ □□□□□ □□□ □□ □□ □□□ □□□□□?

A)

```

[
  {
    "attributes": ...,
    "payload": "100"
  },
  {
    "attributes": ...,
    "payload": "200"
  }
]

```

B)

```

{
  "0": "100",
  "1": "200"
}

```

C)

["100", "200"]

D)

```

{
  "0": {
    "attributes": ...,
    "payload": "100"
  },
  "1": {
    "attributes": ...,
    "payload": "200"
  }
}

```

A. B

B. D

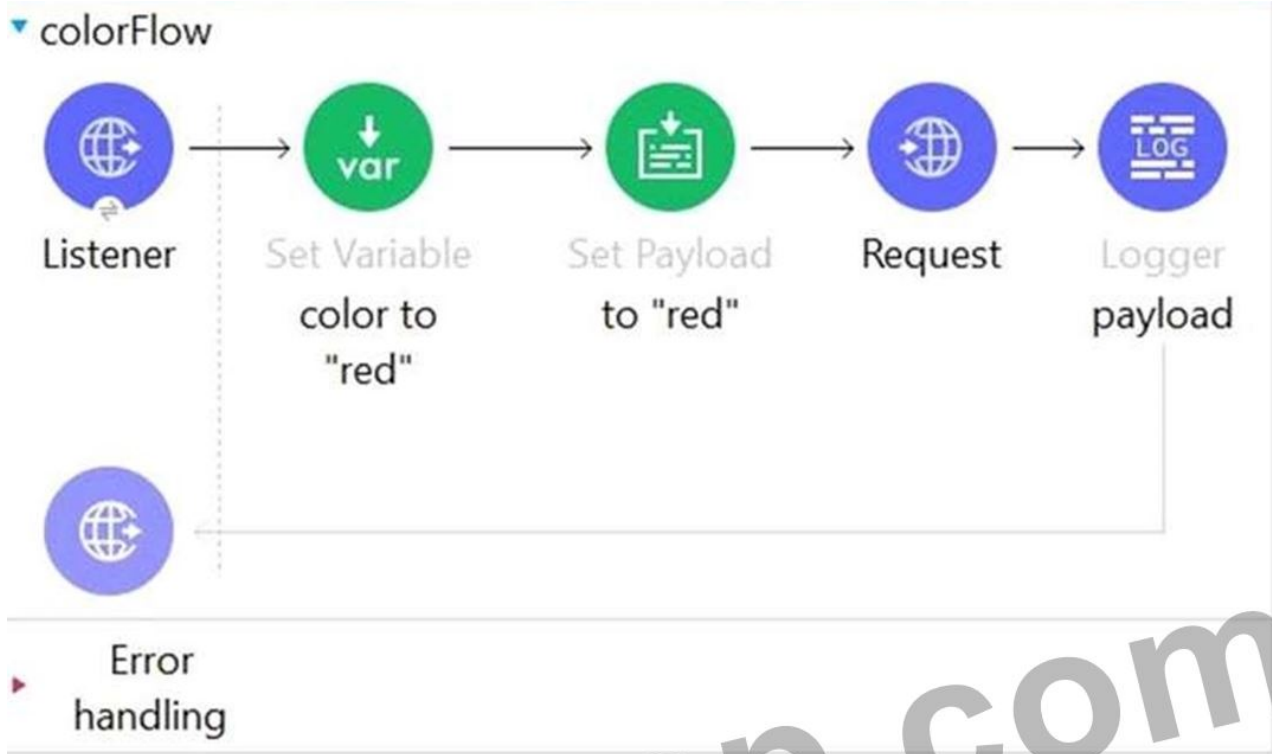
C. C

D. A

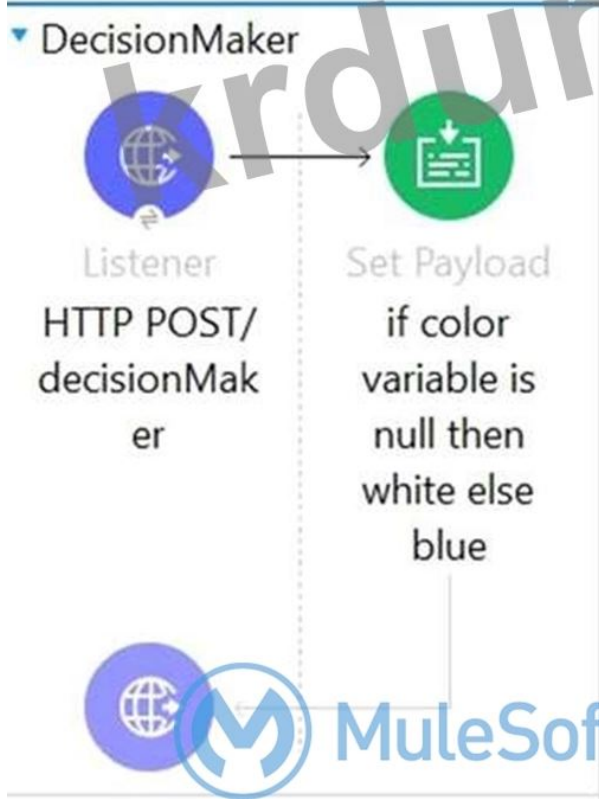
Answer: ([SHOW ANSWER](#))

NEW QUESTION: 89

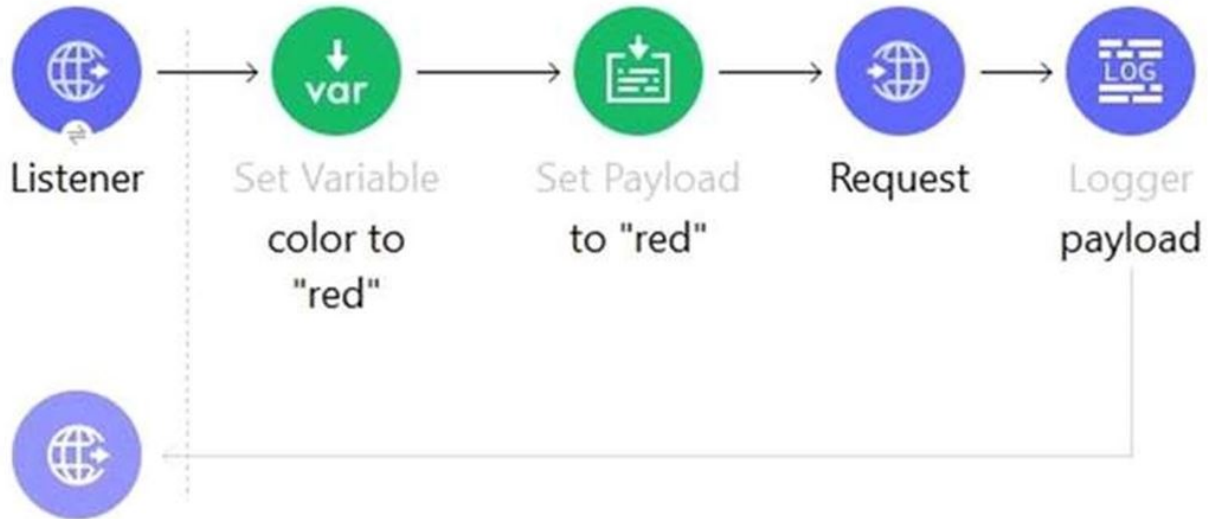
□□□□ □□□□□. □□ □□□□ color □ payload□□ □□□ □□ "□□□"□□ □□□□□. □□ □□ HTTP POST □□□ decisionColor □□□ HTTP □□□□ □□□□□. HTTP □□□ □□□ □ Logger □□ □□□ □□□□ □□ □□□□□?



▶ Error handling



colorFlow



Error handling

DecisionMaker



```

<flow name="colorFlow" doc:id="c0a245d-7f64-41ec-b611-f3d55407a9c1" >
  <http:listener doc:name="Listener" doc:id="e0342e2c-504b-44c0-96e7-b356528215fb" config-ref="HTTP_Listener_config" path="/color"/>
  <set-variable value="red" doc:name="color to 'red'" doc:id="8bc2c51f-b23b-4b5e-0fc4-84c7f4a6ed1f" variableName="color"/>
  <set-payload value="red" doc:name="to 'red'" doc:id="0d7ee5b9-3a8f-461d-b1f4-0e026faf8701" />
  <http:request method="POST" doc:name="Request" doc:id="54725652-f25c-459c-811b-75f551c6d4f5"
  config-ref="HTTP_Request_configuration" path="/decisionmaker"/>
  <set-payload value="payload" doc:name="payload" doc:id="69c768e3-f169-469b-b60e-863f051428e5" message="#[payload]"/>
</flow>

<flow name="DecisionMaker" doc:id="14f2a5e7-84f9-4171-891e-3d0ed16d6d5e" >
  <http:listener doc:name="HTTP_POST/decisionMaker" doc:id="b9199cd2-5106-4bf9-84ca-b5aab42f4532" config-ref="HTTP_Listener_config" path="/decisionmaker"/>
  <set-payload value="#[if (vars.color == null)'white' else 'blue']"
  doc:name="if color variable is null then white else blue" doc:id="9fb8201c-d719-4856-9a56-ed989bf855ce" />
</flow>
  
```

- A.
- B.
- C.

D. □□ □□□

Answer: ([SHOW ANSWER](#))

□□□ □□□□□.

□□□ □□□□ □ □□□ HTTP □□□ □□□□ □□□ □ □□□ □□□□ □□□ □□□ □ □□□ □□□□.

□□□ □□□ □□□ □□ □□□□□.

1) □□ □□□ □□□□□ □□□

2) □□□□□ □□□□□ □□□

3) □□ □□□ HTTP □□□ □□□□ □□□□, □□ □□□ □□□□ □□□ □□□ □ □□□ □□□□□(□□ □□ □□□ □□□ □□□ □□□ □□□□).

4) □□□ □□□□ □□□ □□□□ □□ □□□ null□□□ □□□□□ □□□□ □□□□□.

5) □□ □□□□□ □□□□ □□□□□ □□□□□.

6) □□ □□□ □□□ □□□□□.

NEW QUESTION: 90

API □□□□ □□ API □□ □□□ API □□□□□ Anypoint Exchange□ API □□□ □□□□ □□□□□ □□□□□. API □□□□□ CloudHub□□ □□ □ □□□□ API □□□□□ □ □□□□.

SLA □□ □□□ □ API □□□□□ □□ API □□□□□ □□□□□□□.

□□ API □□□□ API □□□□ □□□□□ □□□ □□□ □□ □□□ □□□□□ ID□ □□□ □□ □□□ □□□□ □□ □ □□□□?

A. Anypoint Exchange□ □□ □□ API □□□□ API □□□□ □□ □□□ □□□□□ □□□□ □□□□

B. Anypoint Studio□□ API □□□ □□ APIkit□□ □□□ □□ □□□□

C. Anypoint Studio□□ Rest Connect□□ □□□ □□□□ □ API □□

D. □□□ □□□□□ □□□ □□□ API □□□□ □□ □□□

Answer: A ([LEAVE A REPLY](#))

* Anypoint Platform□ □□□□□ □□□□□□□ □□□□ □□□□□ ID□ □□□□□ □□□ □□ □□□ □□□□ □□ □□□□□.

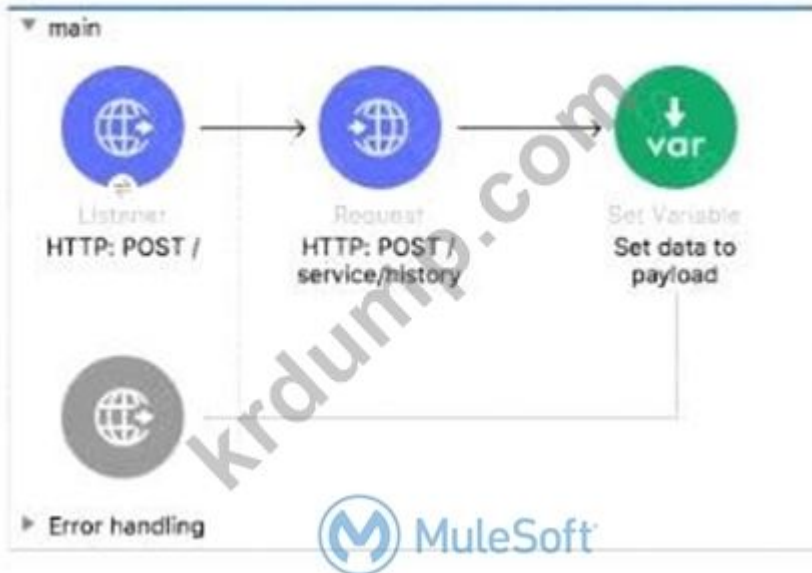
* □□□□□ □□□□□□□ API□ □□ □□□□ □□□□ □□□□□□□ □□ API □□ □□ □ □□□□□.

* □□□□□ ID □□ □□□□ □□□□ API□ □□□ □□□ □□ □□□□□□□□□□ □□□□ □ □□□□.

Anypoint Exchange□ □□ □□ API □□□□

NEW QUESTION: 91

□□□ □□□□□.



□□ □□ □□ □□ □□□□ □□□□ □□ □□□ □□□ □□□ □ □□□□?

- A. Mule □□□ □ □□□ □□
- B. ObjectStore □ □/□ □
- C. Mule □□□ □□ □□□ □□
- D. □□ □□

Answer: B ([LEAVE A REPLY](#))

MCD-Level-1 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ MCD-Level-1 □
 □! DumpTop □ □□ **MCD-Level-1** □□ □□□ □□□□□□, DumpTop MCD-Level-1 □□
 □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
 MCD-Level-1 □□□ □□□□□. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
 (235 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 92

□ □□□□□□ MySQL □□□□□□ □□□□ □□□□□□. □□□□□□ Cloudhub □ □□□ □
 □□□ Anypoint Studio □□ □□□□□□. Cloudhub □ □□□□□ □□□ □□ □□ □□ □□□
 □□□□□ □□□□□ □□ □□□□ □□□ □□□□ □□□□ □□□?

- A. □ □□ □□ □□ 1) □□□□ □□ □□ 2) □□□□ □□ □ □□□ □□
- B. □□ □□ □ □□□ □□□ □□ □□ 1) □□□□ □□ □□ 2) □□□□ □□ □ □□□ □□
- C. □□ □□□ □□ 2) □□□□ □□ □ □□□ □□
- D. □□ □□□ □□ 1) □□□□ □□ □□

Answer: ([SHOW ANSWER](#))

□□□□ □□ □□□ □□□□ Studio □□ □□ □□□ □□□ □□ Mule □□□□□ □□ □□
 □□ □□ □□□□ □ □□□ □□□□□□ □□□ □ □□□□□. □□□□□ JAR □□□□ Studio □
 □ □□□□ □□ □□□ □ □□□□ □□□□ □□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□.

Cloudhub.
Mule
<https://docs.mulesoft.com/studio/7.5/import-export-packages#exporting-a-studio-project-to-a-deployable-mule-application>

NEW QUESTION: 93

.



```

<flow name="main">
  <http:listener doc:name="HTTP: POST /" config-ref="HTTP_listener_config" path="/" />
  <try doc:name="Try" >
    <validation:is-null doc:name="payload" value="#[payload]" message="Validation Error"/>
    <error-handler >
      <on-error-propagate enableNotifications="true" logException="true" doc:name="On Error Propagate">
        <set-payload value="Error - Try scope" doc:name="Error - Try Scope"/>
      </on-error-propagate>
    </error-handler>
  </try>
  <set-payload value="Success - main flow" doc:name="Success - main flow" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue">
      <set-payload value="Error - main flow" doc:name="Error - main flow" />
    </on-error-continue>
  </error-handler>
</flow>

```



Try Catch blocks are used to handle exceptions in a flow. The try block contains the main logic of the flow, and the catch block contains the logic to handle any exceptions that occur. In this example, the try block contains a validation step that checks if the payload is null. If the payload is null, an exception is thrown with the message "Validation Error". The catch block contains an on-error-propagate element that sets the payload to "Error - Try scope".

- A. Success - main flow
- B. Error - Try scope
- C. Error - main flow
- D. Success - Finished Main Flow

Answer: B (LEAVE A REPLY)

The correct answer is B. Error - Try scope. This is because the validation step in the try block checks if the payload is null. If the payload is null, an exception is thrown with the message "Validation Error". The catch block contains an on-error-propagate element that sets the payload to "Error - Try scope".

The correct answer is B. Error - Try scope.

- 1) HTTP listener doc:name="HTTP: POST /" config-ref="HTTP_listener_config" path="/" />
- 2) try doc:name="Try" >
- 3) Is Number [error.description] = "[error.description] INTEGER [error.errorType] = VALIDATION:INVALID_NUMBER
- 4) On Error Continue [error.description] "Error - Sub Flow" [error.description]
- 5) "Error - Sub Flow" [error.description] [error.description] "Success - Finished Main Flow" [error.description]
- 6) "Success - Main Flow" HTTP [error.description] [error.description] HTTP [error.description]: 200 [error.description] Mule Message [error.description] (RED in, GREEN out) [error.description] on error continue [error.description] mainFlow [error.description] childFlow [error.description] mainFlow [error.description]

#[error.description] = "[error.description] INTEGER [error.errorType] = VALIDATION:INVALID_NUMBER"

#[error.errorType] = VALIDATION:INVALID_NUMBER

- 4) On Error Continue [error.description] "Error - Sub Flow" [error.description]
- 5) "Error - Sub Flow" [error.description] [error.description] "Success - Finished Main Flow" [error.description]
- 6) "Success - Main Flow" HTTP [error.description] [error.description] HTTP [error.description]: 200 [error.description] Mule Message [error.description] (RED in, GREEN out) [error.description] on error continue [error.description] mainFlow [error.description] childFlow [error.description] mainFlow [error.description]

- A. Mule 4
- B.
- C.
- D.

Answer: C (LEAVE A REPLY)

DataWeave.

Mule 4.

NEW QUESTION: 95

The image shows two MuleSoft flow diagrams and their corresponding XML code. The 'main' flow starts with an HTTP listener for GET requests, followed by a flow reference to a 'private' flow, and then a 'Set Payload' action for success. An error handler is attached to the flow reference, which contains an 'On Error Continue' action that sets the payload to 'Error - main flow'. The 'private' flow starts with a 'Source' connector, followed by an 'is-number' validation. An error handler is attached to the validation, containing an 'On Error Continue' action that sets the payload to 'Error - private flow'.

```

<flow name="main" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_listener_config" path="/" />
  <flow-ref doc:name="private" name="private"/>
  <set-payload value="Success - main flow" doc:name="" />
  <error-handler>
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - main flow" doc:name="" />
    </on-error-continue>
  </error-handler>
</flow>

<flow name="private" >
  <validation:is-number numberType="INTEGER" doc:name="payload" value="#[payload]" message="Validation Error" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - private flow" doc:name="" />
    </on-error-continue>
  </error-handler>
</flow>

```

HTTP

- A. -
- B. -
- C. -

D. □□ □□

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 96

□□□□ □□□□□.

```

{
  "user": "vivek.singh@acme.com",
  "items": [
    {"item": "bike", "price": 120.45, "qty": 1}
  ]
}

```



□□□ □□ □ □□ □□□ □□□□ □□ □□□ □□□□□ □□□ □ □□□□□
 POST □□□ □□□□ □□ Mule □□□□□□□ □□ □□□□. Shipping □□□□ □□ □□□
 □ □□□ □□□ □□ □□ □□□ □□□□□. Shipping □□□□ Shipping Address □□□
 ShippingAddress□□ □□□□ □□□□□. .
 Set Payload □□□□ □□ □□□ □□□□□ □□ □□ □□□ □□ □□ □□□□
 ShippingAddress □□□ □□□ □□□ shippingInfo □□ □□□□ □□□. □□□ □□□□ Set
 Payload □□□□ □□ □□□□ □□□ □□□ □□□□□?

A)

```
{
  items: attributes.shippingAddress.items
  shippingInfo: payload
}
```

B)

```
{
  items: payload.items,
  shippingInfo: vars.shippingAddress
}
```

C)

```
{
  items: payload.items,
  shippingInfo: shippingAddress
}
```

D)

```
{
  items: vars.shippingAddress.items
  shippingInfo: payload
}
```


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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 97

□□□□ □□□□□. □□□□ API□□ □□□ □ Book □□□ □□□ Book □□□ □□□□□
□. □ □ □□□ □□□ □ □□□ □□□□ API□ □□□ RAML□ □□□□□?

#%RAML 1.0 DataType	#%RAML 1.0 NamedExample
# bookDataType.raml	# bookExample.raml
<pre> type: object properties: ID?: integer title: string author: string publisher?: string year: integer ISBN: type: string required: true </pre>	<pre> bookExample: ID: 101 title: Shakespeare author: Encyclopaedia Britannica publisher: John Wiley & Sons year: 2007 ISBN: "0471767840" </pre>



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

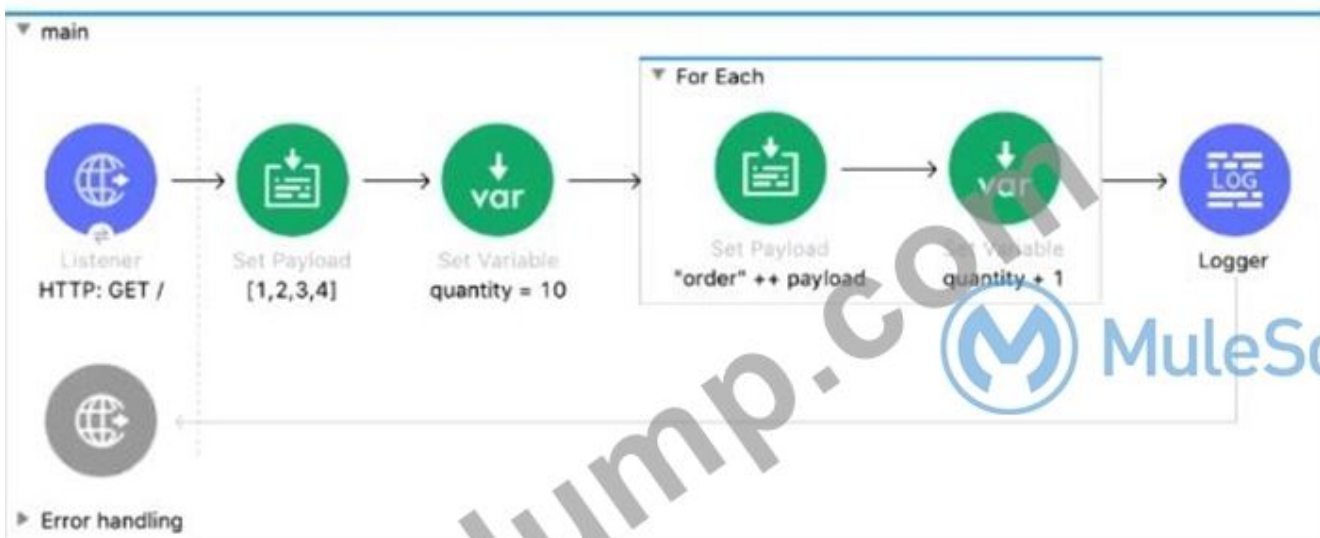
/books:
  post:
    body:
      application/json:
        type: Book
        examples:
          inputs: !include BookExample.raml
        responses:
          201:
            body:
              application/json:
                example:
                  message: Book added
  
```

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 98

.



```

<flow name="main" >
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_listener_config" path="/" allowedMethods="GET"/>
  <set-payload value="#[[1,2,3,4]]" doc:name="[1,2,3,4]" />
  <set-variable value='10' doc:name="quantity = 10" variableName="quantity" />
  <foreach doc:name="For Each" >
    <set-payload value='#["order" ++ payload]' doc:name="'order" ++ payload' />
    <set-variable value="#[vars.quantity + 1]" doc:name="quantity + 1" variableName="quantity" />
  </foreach>
  <logger level="INFO" doc:name="Logger" message='#[[ payload, vars.quantity ]]' />
</flow>

```

?

- A. [[1,2,3,4], 14]
- B. [234, 14]
- C. [[1, 2, 3, 4], 14]
- D. [[1,2,3,4], 10]



NEW QUESTION: 103

API CloudHub vCore Mule API Manager VPC API Manager JAR API Manager API Manager

- A. API Manager Runtime Manager API
- B. VPC API Manager VPC
- C. Mule JAR API Manager API
- D. API Manager API

Answer: D (LEAVE A REPLY)

NEW QUESTION: 104

additem DataWeave



additem DataWeave

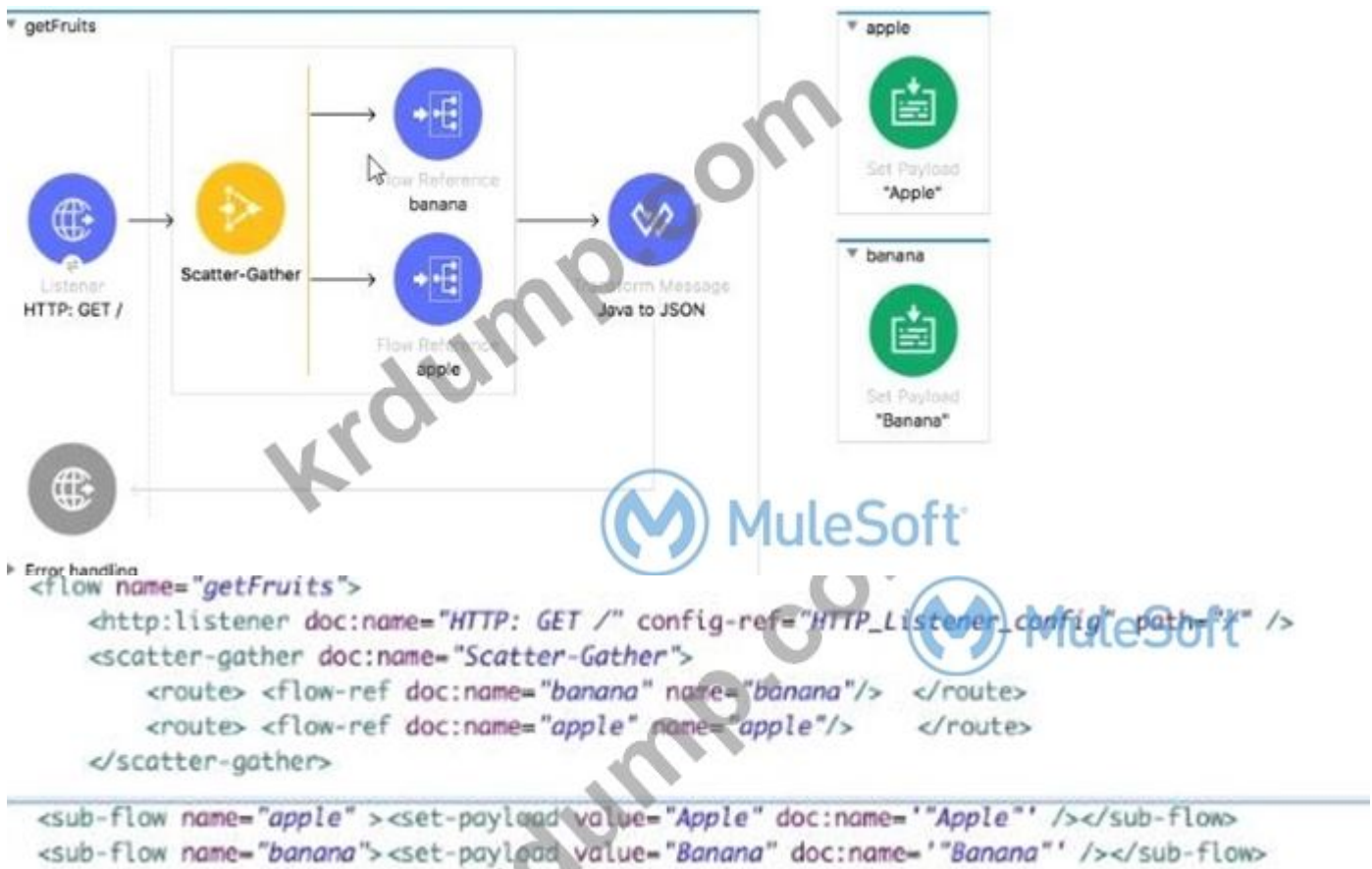
addItem □□ □□□ □□□□ □□□ 100□ □□□ □□□□ □□□ □□□□ □□ createOrder
 □□□□ Set Payload □□□□ □□ □□□ DataWeave □□□ □□□□□?

- A. lookupf "addItem", { □□□□: { price: "100", item: "router", itemType: "cable" } })
- B. lookupf "addItem", { price: "100", item: "router", itemType: "cable" })
- C. addItemf { □□□□: { price: "100", item: "router", itemType: "cable" } >)
- D. addItemf { price: "100", item: "router", itemType: "cable" })

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 105

□□□□ □□□□□.



□ □□□□□□ http://localhQst:8081□ □□□ □□□□□. □□□ □□□ □□□□□ □□□ □
 □□□□?

A)
 ['Banana', 'Apple']

B)
 {
 "0": "Banana",
 "1": "Apple"
 }

C)

□□ □□

B. □□□□

□□ □□ □□□□

C. □□□□

□□ var □□ □□ □□□□

D. □□□□

Answer: D ([LEAVE A REPLY](#))

MCD-Level-1 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ MCD-Level-1 □
 □! DumpTop □ □□ **MCD-Level-1** □□ □□□ □□□□□□, DumpTop MCD-Level-1 □□
 □□□ □□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
 MCD-Level-1 □□□ □□□□□□. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
 (235 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 107

RAML □□□ □ □□ □□□□ □□ □□ □□□□ □□□ □□□□□ □□□□□□.
 □□ ID 1234□ □□□ □□□□ □□□□ □□□□□ □□ MuleSoft□ □□□□ URI□ □□□□
 □?

- A. /customers?operation=get&custid=1234
- B. /customers/1234
- C. /customers/custid=1234
- D. /customers?custid=true&custid=1234

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 108

Utility.dwl □□□ src/main/resources/modules□ Mule □□□□□ □□□□□. Utility.dwl□ □□□
 □ □□□ □□□□ □□ □□□□ pascalize□□ □□□ □□□□□□.
 □□ □□□ □□ □□□□ pascalize □□□ □□□□ □□□□ DataWeave□ □□□□□?

A)

```

@dw 2.0
output application/json
import modules::Utility
---
pascalize( "max mule" )

```

B)

```

@dw 2.0
output application/json
import modules:::utility
---
pascalize( "max mule" )

```

C)

```
dw 2.0
output application/json
import modules::Utility
---
Utility::pascalize( "max mule" )
```

D)

```
dw 2.0
output application/json
import modules.Utility
---
Utility.pascalize( "max mule" )
```

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

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 109

dataweave `{ db:port }` ?

- A. { db : p('db.port')}
- B. { db : {db:port}}
- C. { db : p['db.port']}
- D. Dataweave `{ db:port }` ?

Answer: A ([LEAVE A REPLY](#))

dataweave `{ db:port }` ?

NEW QUESTION: 110

dataweave `{ db:port }` ?


```

id POST . POST .
id POST .
{
  "GoerdiLa Forge",
  "1 Westland CA",
  "customer_since": "2014-01-04",
  "4829.29",
  "bank_agend_id": "12556"
}

```

NEW QUESTION: 111

Mule `loginUser` `DataWeave` `WebStore` `dvA` `src/main/resources/libs/dw` `WebStore.dwl` `DataWeave` `loginUser` `"cindy.park@example.com"` `loginUser` `loginUser`?

A)

```

import libs.dw
---
WebStore.loginUser( "cindy.park@example.com" )

```

B)

```

import * from libs::dw
---
WebStore::loginUser( "cindy.park@example.com" )

```

C)

```

import libs.dw.WebStore
---
loginUser( "cindy.park@example.com" )
import * from libs::dw::WebStore
---
loginUser( "cindy.park@example.com" )

```

A. B

B. C

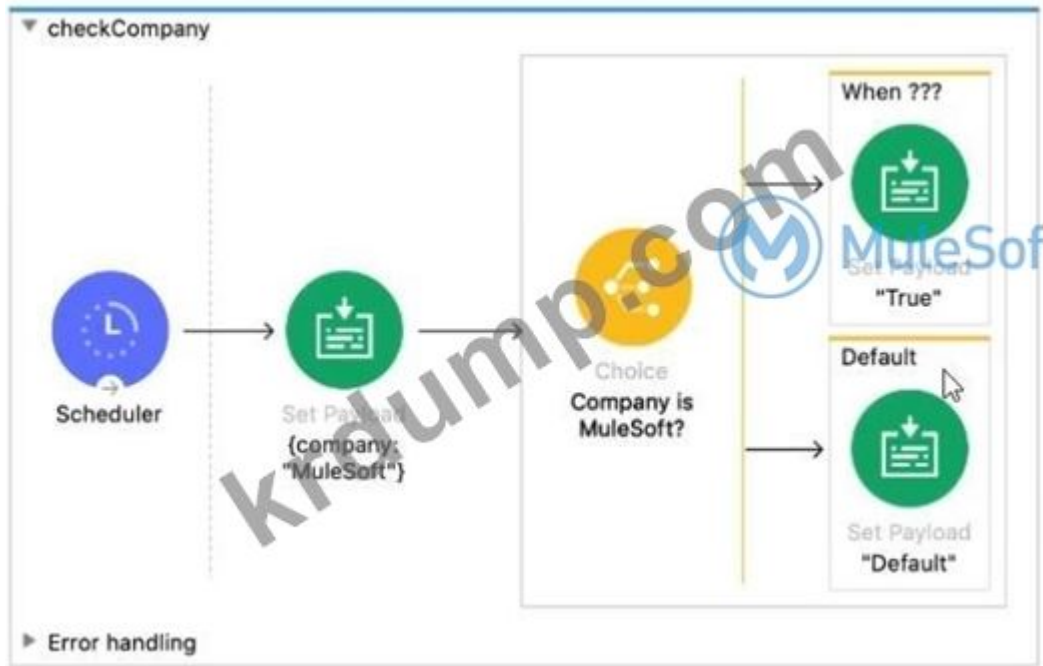
C. D

D. A

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 112

□□□□ □□□□.



```

<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>

```

Choice □□□□ □□ <when> □□□□ □□□□ □□□.

Mule □□□□ □□□ □□ □□□□ □□□□□ □□□ <□> □□□□ □□□□□□?

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

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 113

□□□□ □□□□.



Which of the following is the correct content type for the XML body of the POST request to the ACME Order API? Select two.

Which of the following is the correct content type for the XML body of the POST request to the ACME Order API?

- A. application/octet-stream
- B. application/xml
- C. application/xml
- D. application/octet-stream

Answer: C (LEAVE A REPLY)

NEW QUESTION: 114

Which of the following is the correct choice for the route configuration?



Which of the following is the correct choice for the route configuration? Select two.

- A. 1, 2
- B. 1, 2, 3

C. □□ 1

D. □□1, □□2

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 115

□□□□ □□□□□.



□□□□ □□ □□□ addItem □□ □□□□ DataWeave □ □□□□ □□ □□□ □□□□.

addItem □□ □□□ □□□□ □□□ 100□ □□□ □□□□ □□□ □□□□ □□ createOrder

□□□□ Set Payload □□□□ □□ □□□□ DataWeave □□□ □□□□□?

- A. addItem { □□: "100", □□: "□□□", □□ □□: "□□□" }
- B. lookupf "addItem", { □□□□: { price: "100", item: "router", itemType: "cable" } })
- C. addItem { □□□□: { price: "100", item: "router", itemType: "cable" } >)
- D. lookupf "addItem", { price: "100", item: "router", itemType: "cable" })

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 116

□□□ □□□□□.



Mule `tracing.host` HTTP `tracing.host`. Mule `config.yaml` `tracing.host`.
Mule `config.yaml` HTTP `tracing.host` `tracing.host` `tracing.host` `tracing.host`?

- A. `tracing.host`
- B. `tracing.host`
- C. `tracing.host`
- D. `tracing.host`

Answer: (SHOW ANSWER)

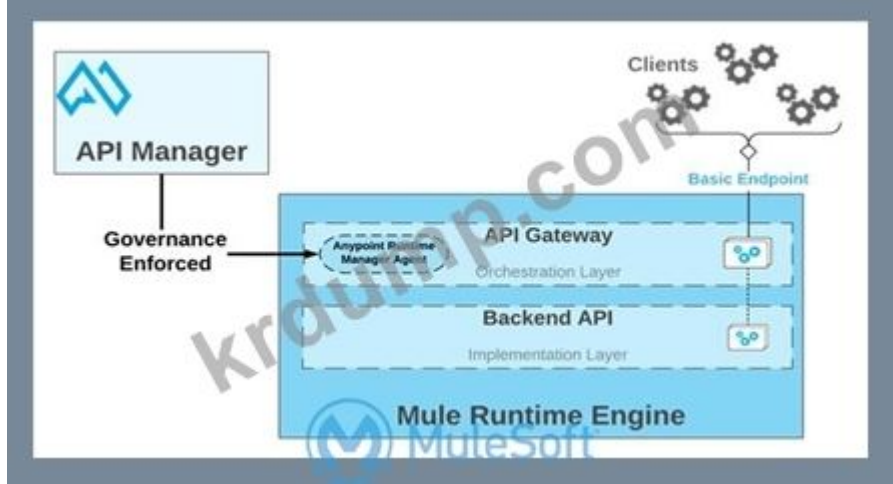
NEW QUESTION: 118

API Gateway `tracing.host`?
A. API `tracing.host` `tracing.host` `tracing.host`.
B. `tracing.host` `tracing.host`.
C. `tracing.host`, `tracing.host` `tracing.host` `tracing.host`.
D. `tracing.host`, `tracing.host` `tracing.host` `tracing.host`

Answer: D (LEAVE A REPLY)

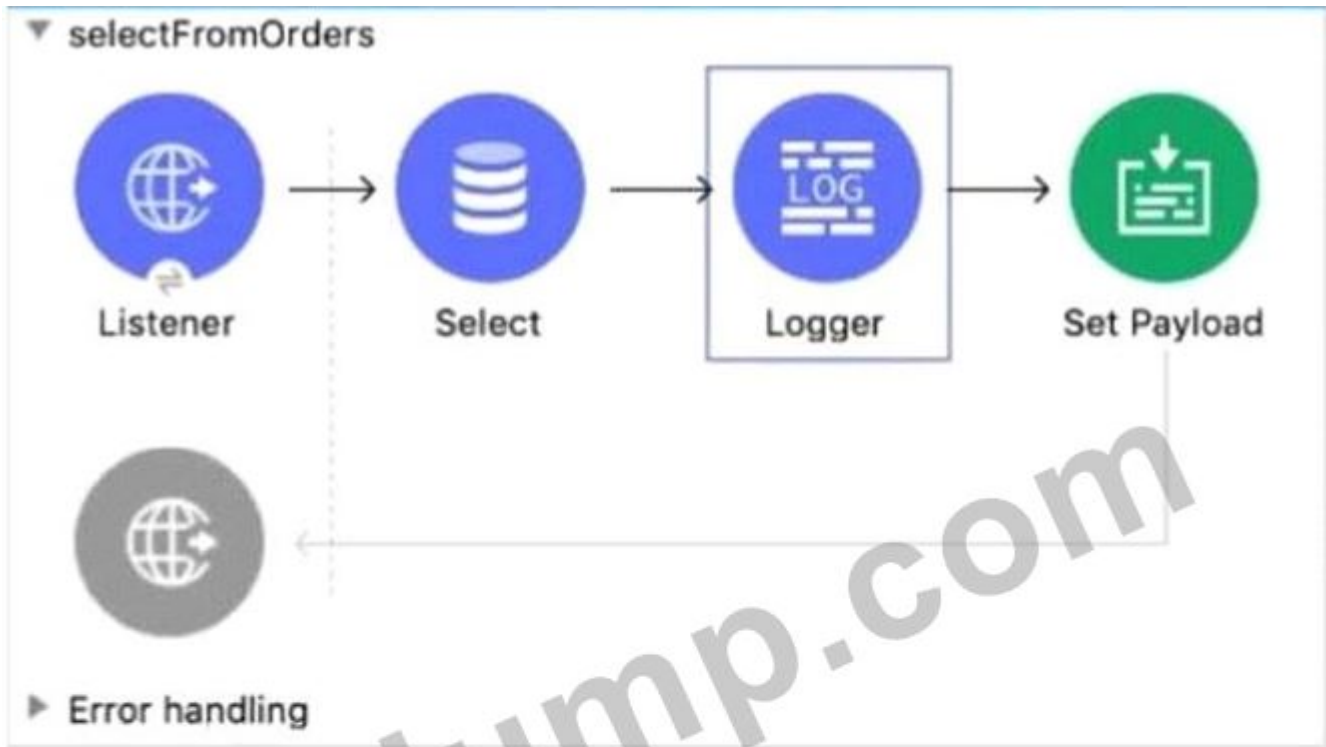
`tracing.host`, `tracing.host` `tracing.host`.
MuleSoft `tracing.host`: <https://docs.mulesoft.com/api-manager/2.x/api-gateway-capabilities-mule4>

- API Gateway `tracing.host` `tracing.host` `tracing.host`.
- 1) `tracing.host` `tracing.host` `tracing.host`
 - 2) `tracing.host` `tracing.host`
 - 3) `tracing.host` `tracing.host`
 - 4) `tracing.host` `tracing.host` `tracing.host` (`tracing.host` `tracing.host`. API Manager `tracing.host` `tracing.host`)



NEW QUESTION: 119

`tracing.host` `tracing.host`. `tracing.host` `tracing.host` `tracing.host` `tracing.host`?



- A. □□□
- B. □□
- C. □□
- D. □□

Answer: C (LEAVE A REPLY)

□□□□□□□□ □□ □□ □□□ □□□□□□.
 □□□ □□□□□

NEW QUESTION: 120

□□ □□□□□□□□ □□ □□□□ SLA □□ □□□ □□□ □□/□□/□□□□ □ □□□□□?

- A. API □□
- B. API □□
- C. API □□□□□

D. API □□□

Answer: D (LEAVE A REPLY)

□□ □□□□ API Manager□ □□□□ SLA □□ □□□ □□□ □□/□□/□□□ □ □□□□.

NEW QUESTION: 121

□□□□ □□□□□.

mulesoft □□□□□ □ □□□ □□□□ □□ □□□□□□ □□□ □□□□ □□□ □□□□□ □□□□?



```

org.mule.runtime.api.connection.ConnectionException: Could not obtain connection from data source
used by: org.mule.extension.db.api.exception.connection.ConnectionCreationException: Could not obtain connection from data source
used by: org.mule.runtime.extension.api.exception.ModuleException: java.sql.SQLException: Error trying to load driver: com.mysql.jdbc.Driver : Cannot lo
Class 'com.mysql.jdbc.Driver' has no package mapping for region 'domain/default/app/mule_app',
Cannot load class 'com.mysql.jdbc.Driver': [
Class 'com.mysql.jdbc.Driver' has no package mapping for region '/domain/default',
Class 'com.mysql.jdbc.Driver' not found in classloader for artifact 'container'.]
used by: java.sql.SQLException: Error trying to load driver: com.mysql.jdbc.Driver : Cannot load class 'com.mysql.jdbc.Driver': [
Class 'com.mysql.jdbc.Driver' has no package mapping for region 'domain/default/app/mule_app',
Cannot load class 'com.mysql.jdbc.Driver': [
Class 'com.mysql.jdbc.Driver' has no package mapping for region '/domain/default',
Class 'com.mysql.jdbc.Driver' not found in classloader for artifact 'container'.]
at org.mule.extension.db.internal.domain.connection.JdbcConnectionFactory.createConnection(JdbcConnectionFactory.java:57) ~[mule-db-connector-1.9.
at org.mule.extension.db.internal.domain.connection.DbConnectionProvider.connect(DbConnectionProvider.java:139) ~[mule-db-connector-1.9.3-mule-plu
at org.mule.extension.db.internal.domain.connection.DbConnectionProvider.connect(DbConnectionProvider.java:71) ~[mule-db-connector-1.9.3-mule-plu
at org.mule.runtime.module.extension.internal.runtime.config.ClassLoaderConnectionProviderWrapper.connect(ClassLoaderConnectionProviderWrapper.jav
at org.mule.runtime.core.internal.connection.ConnectionUtils.connect(ConnectionUtils.java:49) ~[?:?]
at org.mule.runtime.core.internal.connection.AbstractConnectionProviderWrapper.connect(AbstractConnectionProviderWrapper.java:64) ~[?:?]
at org.mule.runtime.core.internal.connection.ErrorTypeHandlerConnectionProviderWrapper.connect(ErrorTypeHandlerConnectionProviderWrapper.java:64)
at org.mule.runtime.core.internal.connection.ConnectionUtils.connect(ConnectionUtils.java:49) ~[?:?]
at org.mule.runtime.core.internal.connection.AbstractConnectionProviderWrapper.connect(AbstractConnectionProviderWrapper.java:64) ~[?:?]
at org.mule.runtime.core.internal.connection.DefaultConnectionProviderWrapper.connect(DefaultConnectionProviderWrapper.java:53) ~[?:?]

```

- A. □□□ □□□ URL □□
- B. □□□ □□□□□□ □□ □□
- C. □□□ □□□ □□ □□
- D. □□□ JDBC □□□□ □□

Answer: D (LEAVE A REPLY)

□□□ □□□ □□ □□ □□□□ □□□□ □□□ JDBC □□□□ □□□□□. □□:

java.sql.SQLException: □□□□ □□ □□ □ □□: com.mysql.jdbc.Driver:

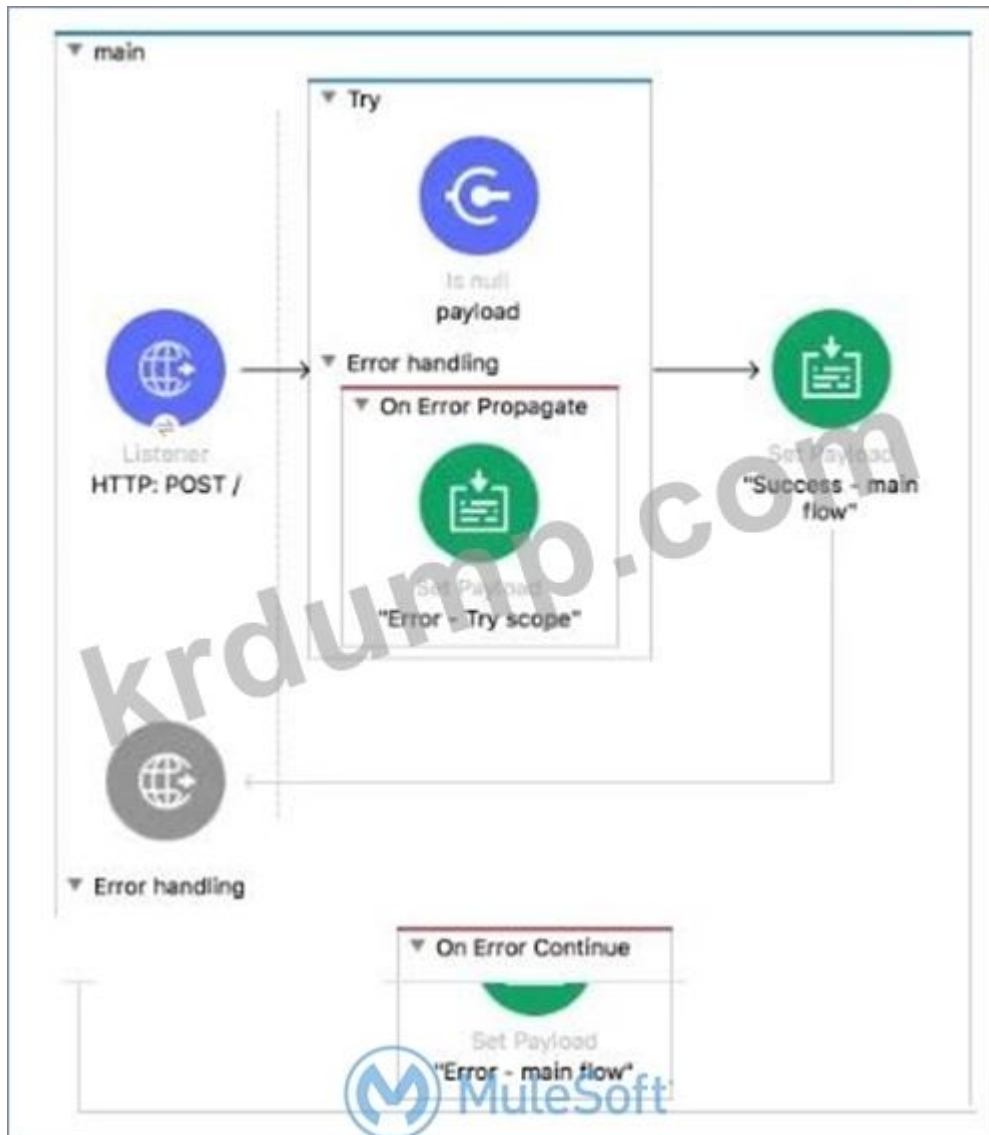
'com.mysql.jdbc.Driver' □□□□ □□□ □ □□□□. ['com.mysql.jdbc.Driver' □□□□□

'domain/default/app/mule_app' □□□ □□ □□□ □□□ □□□□□., 'com.mysql.jdbc.Driver' □ □□□ □□□ □ □□□□.

MCD-Level-1 ☐☐ ☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop ☐☐ ☐☐☐☐ ☐☐☐ MCD-Level-1 ☐
☐! DumpTop ☐ ☐☐ MCD-Level-1 ☐☐ ☐☐☐ ☐☐☐☐☐☐, DumpTop MCD-Level-1 ☐☐
☐☐☐ ☐☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐. ☐☐☐☐☐ ☐☐☐☐ ☐☐☐☐☐ ☐☐ DumpTop
MCD-Level-1 ☐☐☐ ☐☐☐☐☐. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
(235 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 122

☐☐☐☐ ☐☐☐☐☐.



```

<flow name="main">
  <http:listener doc:name="HTTP: POST /" config-ref="HTTP_listener_config" path="/" />
  <try doc:name="Try" >
    <validation:is-null doc:name="payload" value="#[payload]" message="Validation Error"/>
    <error-handler >
      <on-error-propagate enableNotifications="true" logException="true" doc:name="On Error Propagate">
        <set-payload value="Error - Try scope" doc:name="Error - Try scope"/>
      </on-error-propagate>
    </error-handler>
  </try>
  <set-payload value="Success - main flow" doc:name="Success - main flow" />
  <error-handler >
    <on-error-continue enableNotifications="true" logException="true" doc:name="On Error Continue" >
      <set-payload value="Error - main flow" doc:name="Error - main flow" />
    </on-error-continue>
  </error-handler>
</flow>

```

Try □□□ □□□ □□ □□ □□□□ □□□ □□□□□.

□□ □□□□ HTTP □□□□ □□ □□□□□ □□□ □□ □□ □□□□ □□□□□?

Try □□□ □□□ □□ □□ □□□□ □□□ □□□□□. □□ □□□□ HTTP □□□□ □□ □□ □□□□ □□□ □□ □□ □□□□ □□□□□?

- A. □□ - □□ □□
- B. □□ - □□ □□
- C. □□ - □□ □□
- D. □□□ □□ □□

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 123

□□□□ □□□□□.



□ □□□□□□ XML □□□□□ □□ POST □□□ ACME Order API□ □□□□. □□□ □□ □□□.

□□ □□ □□□ □ □□□□□□□ □□□□□ □□□□ □□□ □□□□ □□□?

- A. Content-Type application/octet-stream
- B. Content-Type application/xml
- C. Content-Type applicatron/octet-stream
- D. Content-Type applkation/xml

Answer: B (LEAVE A REPLY)

NEW QUESTION: 124



Choice

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

Answer: C (LEAVE A REPLY)

NEW QUESTION: 125

Mule CloudHub HTTP listener port is defined as `http.port` in the `listener.xml` file.

- A. CloudHub HTTP listener port is defined as `http.port` in the `listener.xml` file.
- B. CloudHub API Manager port is defined as `http.port` in the `listener.xml` file.
- C. MuleSoft HTTP Listener port is defined as `http.port` in the `listener.xml` file.
- D. Mule CloudHub HTTP listener port is defined as `http.port` in the `listener.xml` file.

Answer: (SHOW ANSWER)

CloudHub HTTP listener port is defined as `http.port` in the `listener.xml` file.

MuleSoft □□ □□: https://docs.mulesoft.com/mule-runtime/4.3/deploy-to-cloudhub#prerequisites

NEW QUESTION: 126

□□□ □□□□□.

```

#%RAML 1.0
title: ACME Airlines
version: 1.0

/flight:
  get:
    responses:
      200:
      404:

/airline:
  get:
    queryParameters:
      code: string
    responses:
      200:
      404:

/accounts:
  get:
    responses:
      200:
      404:
  post:
    responses:
      201:
  
```

APIKIT□ RAML □□□□ □ □□ □□ □□□ □□□□□?

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

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 127

MuleSoft□ □□□. Modern API□ □□□ □□ □ □□ □□□□□□?

- A. API □□□ □□ □□ □□ □□ □□
- B. API □□ □□ □ □□ □□□□□□ □□□ □□
- C. □□□ □□□ □ □□□□ □□ □□ □ □□□
- D. API □□□ □□□□□ □□

Answer: B ([LEAVE A REPLY](#))

D.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 130



The screenshot displays the MuleSoft Anypoint Studio interface. On the left, a JSON file named 'list_json_1.json' is open, showing a list of two orders. The first order has an ID of 592, international shipping, and one 'T-shirt Navy' item (size L, price 20). The second order has an ID of 972, domestic shipping, and two 'Cargo Shorts' items (size XL, price 30 each). On the right, the 'Output Payload' is shown as an XML document. It starts with a standard XML declaration, followed by an <order> root element containing two <item> elements. The first item is 'T-shirt Navy' with a total of 20. The second item is 'Cargo Shorts' with a total of 60. A watermark 'krdump.com' and the MuleSoft logo are visible over the image.

A)

```
(
  payload map ( (value, index) ->
    order: {
      item: {
        itemName: value.item,
        total: value.price * value.quantity
      }
    }
  )
)
```

B)

```
order:
  payload map ( (value, index) ->
    item: {
      itemName: value.item,
      total: value.price * value.quantity
    }
  )
```

C)

```
payload map ( (value, index) ->
  order: {
    item: {
      itemName: value.item,
      total: value.price * value.quantity
    }
  }
}
```

D)

```
order:
  ( (
    payload map ( (value, index)
      item: {
        itemName: value.item,
        total: value.price * value.quantity
      }
    )
  )
```

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

Answer: D ([LEAVE A REPLY](#))

Which of the following is the correct syntax for the payload map function? (Select two)

payload map ((value, index) -> { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) -> { item: { itemName: value.item, total: value.price * value.quantity } })

NEW QUESTION: 131

Which of the following is the correct syntax for the payload map function? (Select two)

payload map ((value, index) -> { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) { item: { itemName: value.item, total: value.price * value.quantity } })

payload map ((value, index) -> { item: { itemName: value.item, total: value.price * value.quantity } })

B)

```
{
  "0": "100",
  "1": "200"
}
```

C)

["100", "200"]

D)

```
{
  "0": {
    "attributes": ...,
    "payload": "100"
  },
  "1": {
    "attributes": ...,
    "payload": "200"
  }
}
```

A. A

B. B

C. C

D. D

Answer: D ([LEAVE A REPLY](#))

Scatter-Gather

Mule Mule

```
{
  "0": {
    "attributes": ...,
    "payload": "100"
  },
}
```

```
"1": {  
  "□□": ...,  
  "□□□□": "200"  
}
```

NEW QUESTION: 135

Mule □□□□□□□ □□□□ □□ □□□ □ □□ □□ Cloudhub □□□ □□□ □□□□□?

- A. 0.2 vCore
- B. 0.5 vCore
- C. 1.0 vCore
- D. 0.1 vCore

Answer: D (LEAVE A REPLY)

□□□ 0.1 vCores□□□.

MuleSoft □□ □□: <https://docs.mulesoft.com/runtime-manager/cloudhub-architecture#cloudhub-workers> CloudHub □□□ □□□□ CloudHub□□ □□ □□□□□□□ □□□□ Mule □□□ □□□ □□ □□□□□□□. □□□□ □□□ □□□ □□ □□□ □□□□□□ □□□□ □□□ □ □□□□ □□□ □□ □□□□.

□□□ □□□ □□□, □□□ □ □□□□ □□□ □□□□. □□ □□□ □□□ □□ □ □□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□ □□□□.

Worker Size	Heap Memory	Storage
0.1 vCores	500 MB	8 GB
0.2 vCores	1 GB	8 GB
1 vCore	1.5 GB	12 GB
2 vCores	3.5 GB	40 GB
4 vCores	7.5 GB	88 GB
8 vCores	15 GB	168 GB
16 vCores	32 GB	328 GB

NEW QUESTION: 136

RAML is a REST API specification language. It is used to describe REST APIs. Anypoint Studio provides a visual editor for creating RAML files. Which of the following is a valid RAML snippet?

- A. `resource / {`
- B. `resource / {`
- C. `resource / {`
- D. `resource / {`

Answer: C (LEAVE A REPLY)

`resource / {`
`resource / {`



MCD-Level-1 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ MCD-Level-1 □
□! DumpTop □ □□ **MCD-Level-1** □□ □□□ □□□□□□, DumpTop MCD-Level-1 □□
□□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
MCD-Level-1 □□□ □□□□□. <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
(235 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 137

□□□ □□ □□□□ □□ □ □□ □□□ □□□□ IT □□ □□□ □□□ □□ □□ API□ □□
□□ □□ Mulesoft□ □□ API □□ □□ □□ □□□ □□□ □□□□□□□□.
□□ □□□□ □□□□□□□ □□ □□□□ □□□□ MuleSoft□□ □□□□ API □□ □□ □□
□□□□ □□ API□ □□□ □□□□ □□□□□?

- A. API □□□□□□ □□□ □□□ □□ □ □□□□ □□ □□□□ □□□ □□□ API □□□□ □□□ □□□□ □□□□□.
- B. API □□□□□□ □□□□ □□ □□□□□ □□ □□□ □□□ □ □□□ □□□□□ □□□ □□.
- C. API □□□ □□ □□ □□ □ □□ □□□ □□ □□ □□□ □□□□ □□□□ □□□□□.
- D. API □□□ □□ □□ □□, □□ □□□ □□□□ □ □□ □□□□□ □□□□□□□□.

Answer: B (LEAVE A REPLY)

□□□□ API □□□□□□ □□□□ □□ □□□□□ □□ □□□ □□□ □ □□□ □□□□□ □ □□□□ □□□□.

NEW QUESTION: 138

□□□□□□□ □□ □□□ □□□□ □□ Database SELECT □□□□ □□□□ □□□□□ □ □□□□?

- A. □□
- B. □
- C. □□
- D. □ □□

Answer: D (LEAVE A REPLY)

□□□□ □□ □□□ □ □□□ □□□□□.

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

NEW QUESTION: 139

□□□ □□□□□.





http://localhost.8081/□ □□ □□□□ □□□□ □□□□ □□□□
 □□□□

http://localhost;8081/□ □□ □□□□ □□ □□□□□ □□□□□□?

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 140

Database□□ □□□ □□□□ □□□□ □□□ □□□ □□□□□?

A)

```

fun addKV( object: Object, key: String, value: Any ) =
  object ++ { (key):value }
---
addKV ( {hello: "world"}, "hola","mundo" )

```

B)

```

%function addKV( object: Object, key: String, value: Any ) =
  object ++ { (key):value }
---
addKV ( {hello: "world"}, "hola","mundo" )

```

C)

```
%function addKV( object: Object, key: String, value: Any ) =
  object ++ { (key):value }
---
{ hello: "world" } addKV ( "hola","mundo" )
```

D)

```
fun addKV( object: Object, key: String, value: Any ) =
  object ++ { (key):value }
---
{ hello: "world" } addKV ( "hola","mundo" )
```

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

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 141

API Manager SLA . SLA API ?

- A. API
- B. API .
- C. API API
- D. RAML API

Answer: ([SHOW ANSWER](#))

RAML API MuleSoft . <https://docs.mulesoft.com/api-manager/2.x/tutorial-manage-an-api> .

RAML

SLA API RAML OAS . .

ID .

RAML : .

:

- ID :

:

:

:

client_secret:

:

client-id-required .

/ :

:

is: [JSONPlaceholder ID]

1: JSONPlaceholder ID

2: API Manager SLA

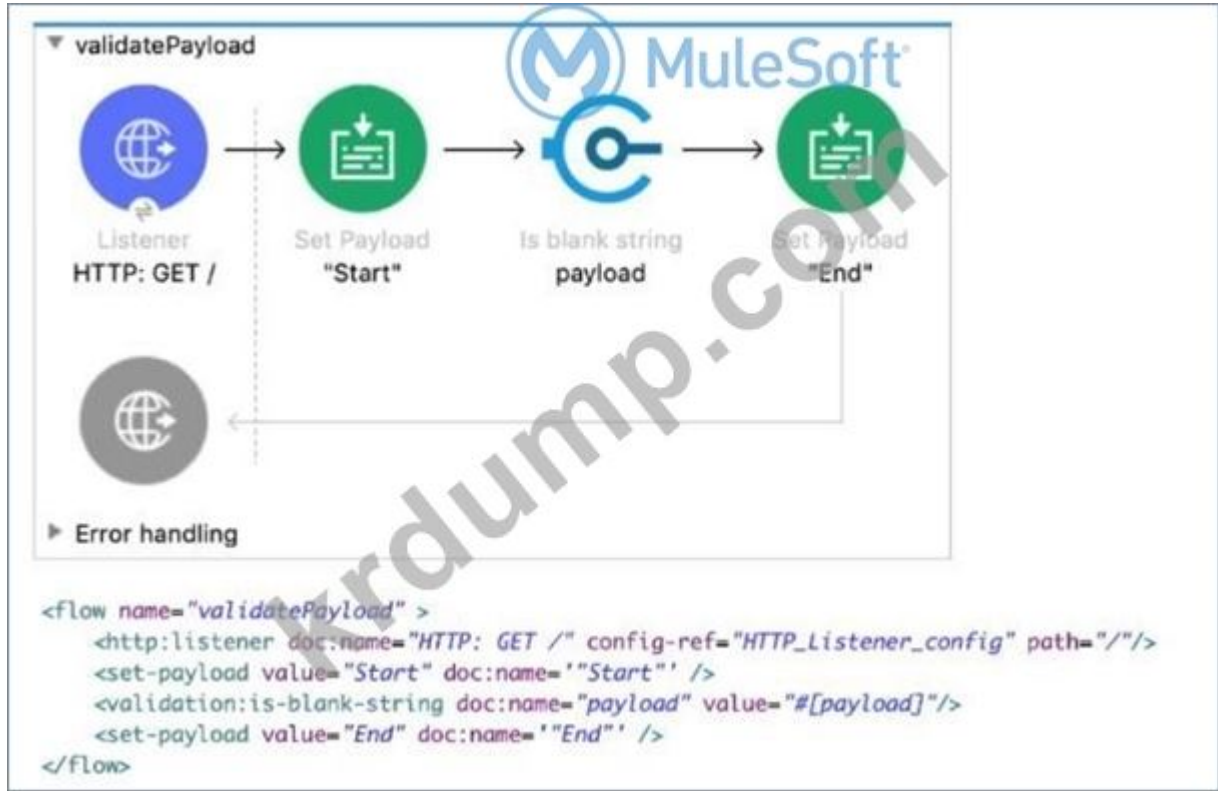
3: API Manager SLA

NEW QUESTION: 142

JSONPlaceholder.

1: JSONPlaceholder HTTP GET

2: JSONPlaceholder ID



- A. ""
- B. "JSONPlaceholder ID"
- C. "ID"
- D. "ID"

Answer: (SHOW ANSWER)

NEW QUESTION: 143

JSONPlaceholder ID Logger ID

- A. #[JSONPlaceholder: " + ID]
- B. ID: \$(ID)
- C. #[JSONPlaceholder: " ++ ID]
- D. ID: #[ID]

Answer: (SHOW ANSWER)

NEW QUESTION: 144

□□□□ □□□ □□ □□ ID□ □□□□ □ □□□□ □□ □□ □□□□ API□ □□□

□. API□ MuleSoft □□ □□□ □□ RAML□ □□□□□□□.

□ □□□□□□ P05555 □□□ □□□□ □□ □□ URI□ □□□□ □□□?

A. /□□/□□=P05555

B. /□□?□□=P05555

C. /□□/□□P05555

D. /□□/{P05555}

Answer: (SHOW ANSWER)

NEW QUESTION: 145

□□□□□ FTP □□□ □□ □□□ □□□□ □□□ □□ □ □□□ □□□ □□□?

A. □□□□ □□□ □□□□□.

B. □□□ □□ □□□ □□□□□.

C. □□□ □□□□ □□ □□□ □□□ □□□□□.

D. □□ □□□□ □□□ □□

Answer: C (LEAVE A REPLY)

FTP □□ □□□ □□□ □ □□□ □□□□□□ □□□□.

MuleSoft □□ □□: <https://docs.mulesoft.com/file-connector/1.3/file-read>

NEW QUESTION: 146

DataWeave □ □□□□ □□ □□□ □□□□□?

A. □□

B. □□

C. □□□

D. □□

Answer: (SHOW ANSWER)

NEW QUESTION: 147

□□□ □□□□□.



□□ □□□□ □□ □□□ □□ □□ □□□ □□□□□.



config.yaml

- A. #[db.username]
- B. #[db:username]
- C. \${db:username}
- D. \${db.username}

Answer: D (LEAVE A REPLY)

NEW QUESTION: 150

APIKit RAML REST APIKit

Mule REST APIKit
 http://localhost:8081/internal http://localhost:8081/external

RAML APIKit

1. raml
2. /
3. :
4. :
5. /:
6. /
7. /
8. /
9. /:

- A. 1
- B. 2
- C. 3
- D. 5

Answer: D (LEAVE A REPLY)

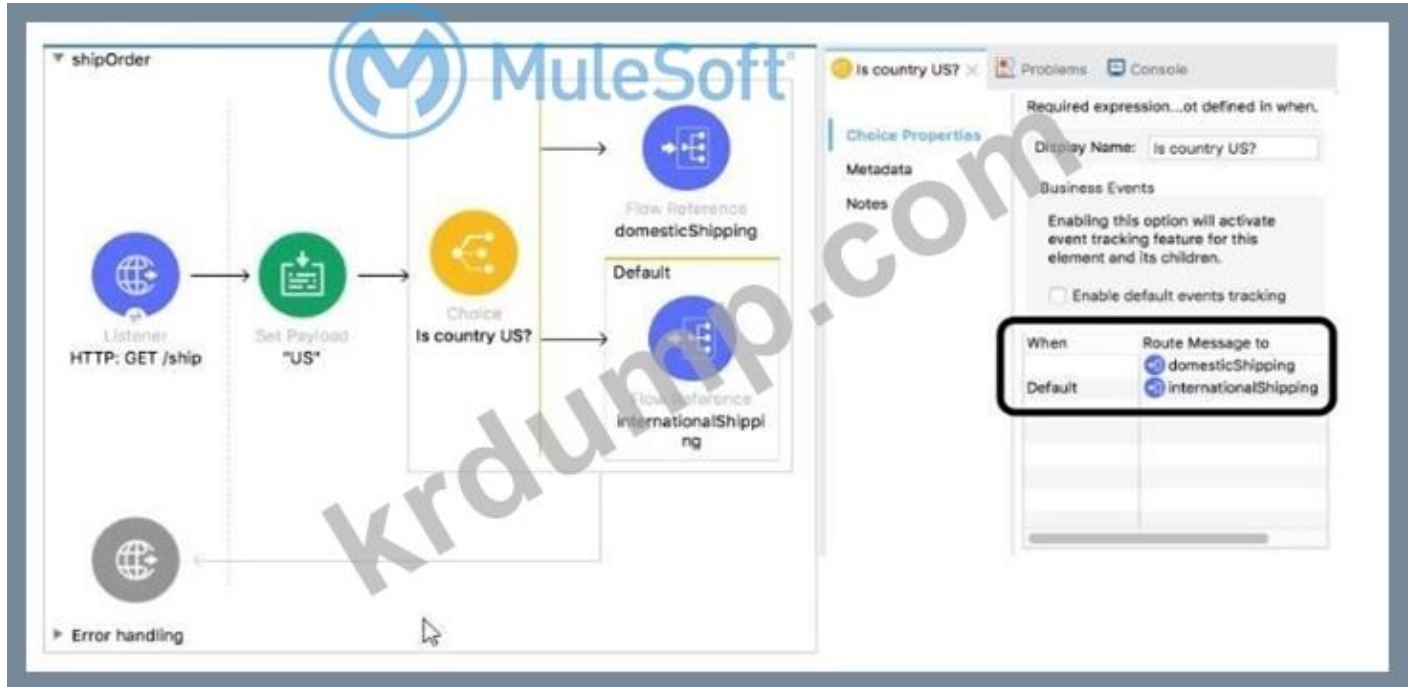
REST APIKit RAML REST APIKit

D. 3

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 154

□□□ □□□□□.



□□□□ domesticShipping □□□□ □□□□□ □□ Choice □□□□ when □□□□ □□ □□□ □□□□ □□□□□?

- A. #[□□□□ == '□□']
- B. #[if(□□□□ = '□□')]
- C. 0#[□□□□ = '□□']
- D. #[if(□□□□ == "□□")]

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 155

□□□ □□□□□.

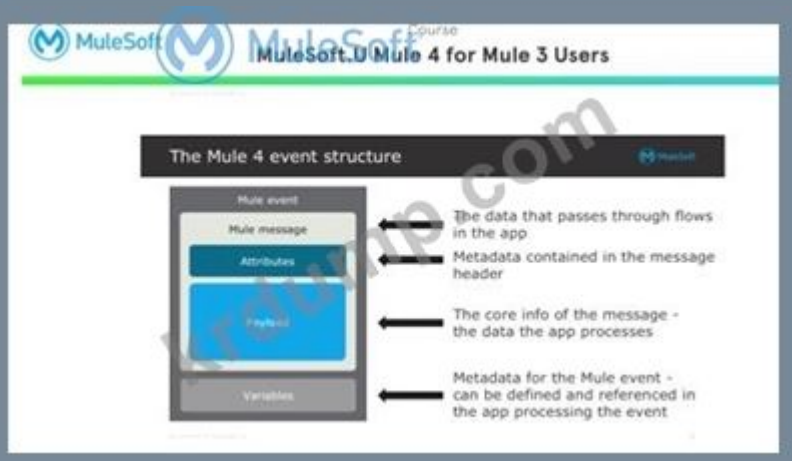
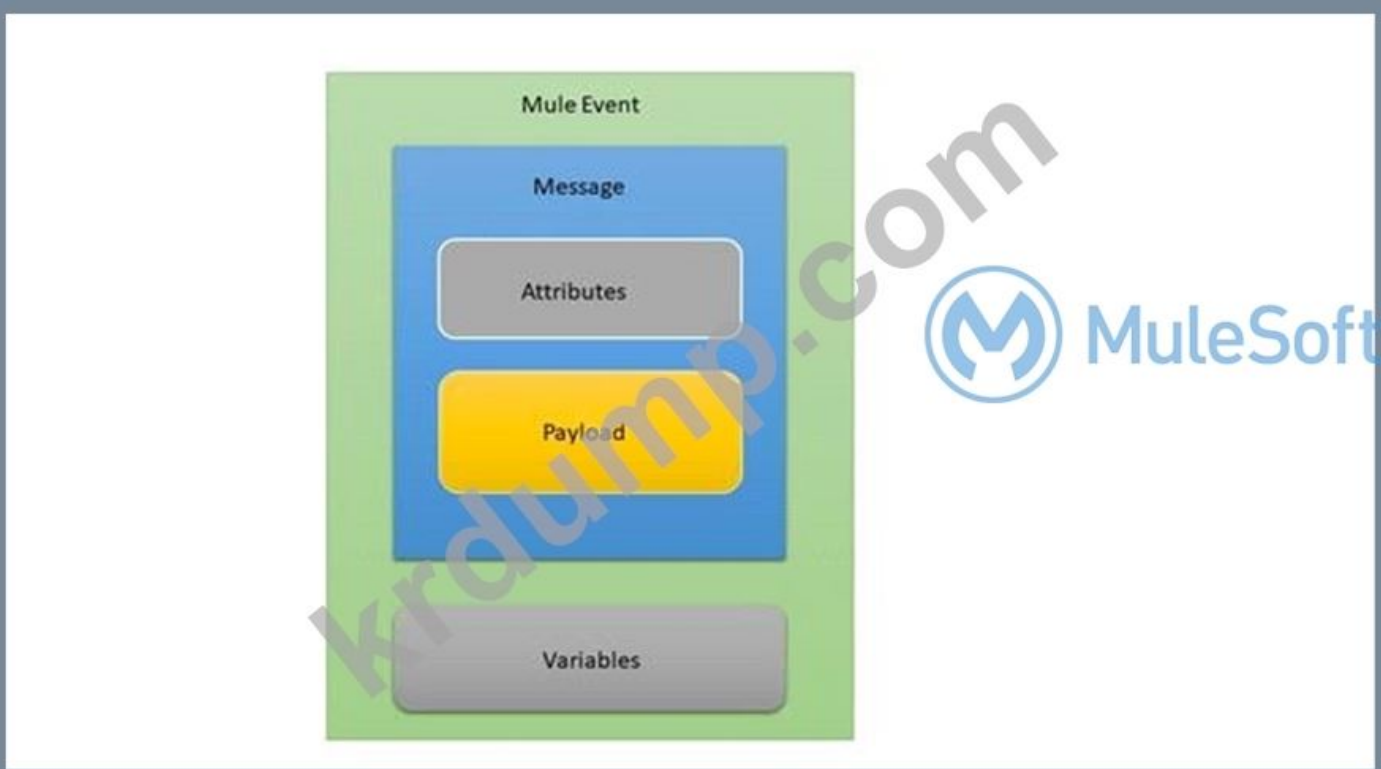


Mule ☐☐☐☐ ☐☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐☐. ☐☐ ☐☐☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐☐?

- A. ☐ ☐☐☐
- B. ☐ ☐☐☐ ☐☐☐☐
- C. ☐ ☐☐☐
- D. ☐ ☐☐☐ ☐☐

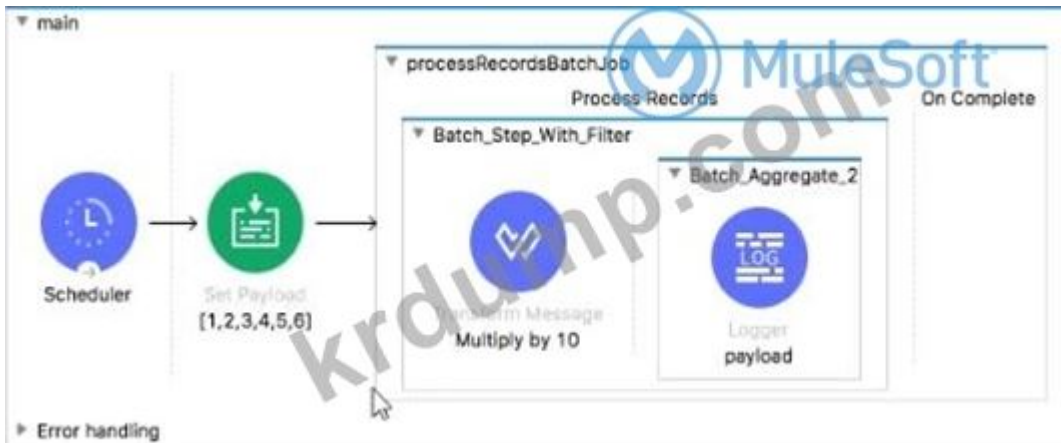
Answer: A ([LEAVE A REPLY](#))

☐☐☐ Mule ☐☐☐☐ ☐☐☐☐☐☐. ☐☐ ☐☐☐☐ ☐☐☐☐☐☐☐☐☐☐.



NEW QUESTION: 156

☐☐☐ ☐☐☐☐☐. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐, ☐☐☐☐☐☐☐☐☐☐☐. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐?



```

<flow name="main" >
  <scheduler doc:name="Scheduler" > <scheduling-strategy >
    <fixed-frequency frequency="10000"/></scheduling-strategy> </scheduler>
  <set-payload value="#[[1,2,3,4,5,6]]" doc:name="[1,2,3,4,5,6]" />
  <batch:job jobName="processRecordsBatchJob" >
    <batch:process-records >
      <batch:step name="Batch_Step_With_Filter" acceptExpression="#[(payload mod 2) = 0]" >
        <ee:transform doc:name="Multiply by 10"><ee:message >
          <ee:set-payload ><![CDATA[%dw 2.0
            output application/java
            payload * 10]]></ee:set-payload>
          </ee:message></ee:transform>
        <batch:aggregator doc:name="Batch_Aggregate_2" size="2">
          <logger level="INFO" doc:name="payload" message="#[payload]"/>
        </batch:aggregator>
      </batch:step>
    </batch:process-records>
  </batch:job>
</flow>

```

- A. [10. 20, 30. 40, 50, 60]
- B. [10. 20] [30, 40] [50, 60]
- C. [20, 40, 60]
- D. [20. 40] [60]

Answer: D (LEAVE A REPLY)

- * $[(payload \text{ mod } 2) = 0]$ = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100.
- * $[(payload \text{ mod } 2) = 0]$ = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100.
- * Aggregator size = 2, so it will process 2 records at a time.
- * $[(payload \text{ mod } 2) = 0]$ = 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100.

[20,40]

[60]

NEW QUESTION: 157

Scenario: A REST API is used to manage temperature data. The API has two endpoints: `updateTemp` and `getTemp`. The `updateTemp` endpoint is used to update the temperature of a location. The `getTemp` endpoint is used to retrieve the temperature of a location. The API is implemented using a REST client. The client is configured to use the following headers: `Content-Type: application/json` and `Accept: application/json`. The client is also configured to use the following authentication headers: `Authorization: Bearer <token>` and `Cookie: <cookie>`. The client is configured to use the following request body for the `updateTemp` endpoint: `{ "location": "New York", "temp": 75 }`. The client is configured to use the following request body for the `getTemp` endpoint: `{ "location": "New York" }`. The client is configured to use the following response body for the `updateTemp` endpoint: `{ "location": "New York", "temp": 75 }`. The client is configured to use the following response body for the `getTemp` endpoint: `{ "location": "New York", "temp": 75 }`.

getTemp □□□ HTTP □□□□ □□ □□□□ □□ □□□ □□□□□?



```
<http:request-config name="HTTP_Request_configuration" doc:name="HTTP Request configuration" >
  <http:request-connection host="localhost" port="8081" />
</http:request-config>
<flow name="test" >
  <http:listener doc:name="HTTP: GET /test" config-ref="HTTP_Listener_config" path="test"/>
  <set-payload value="#[output application/json
[ 70,65,100,60,85 ]]" doc:name="["
  70,
  65,
  100,
  60,
  85
]" />
  <foreach doc:name="For Each" collection="payload">
    <http:request method="POST" doc:name="HTTP: POST /updateTemp" path="/updateTemp"
    config-ref="HTTP_Request_configuration"/>
  </foreach>
</flow>
<flow name="updateTemp" >
  <http:listener doc:name="HTTP POST /updateTemp" config-ref="HTTP_Listener_config" path="updateTemp"/>
  <os:store doc:name="Store payload in temp key in Object Store" key="temp" failOnNullValue="false"/>
</flow>
<flow name="getTemp" >
  <http:listener doc:name="HTTP: GET /getTemp" config-ref="HTTP_Listener_config" path="getTemp"/>
  <os:retrieve-all doc:name="Retrieve all"/>
  <set-payload value="#[output application/json --- payload]" doc:name="output application/json --- payload" />
  <logger level="INFO" />
</flow>
```

A)

```
{
  "temp": [70,65,100,60,85]
}
```

B)

```
{
  "temp": "100"
}
```

C)

```
{
  "temp": "85"
}
```

D)

```
{
  "temp": 70,
  "temp": 65,
  "temp": 100,
  "temp": 60,
  "temp": 85
}
```

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

Answer: (SHOW ANSWER)



□□ □□□ □□ □ □□ □□□□□ □□□ □□□□□?

- A. [□□□□2□□□3□□4, 14]
- B. [[1,2,3,4], 14]
- C. [[□□1, □□2, □□3, □□4], 14]
- D. [[1,2,3,4], 10]

Answer: (SHOW ANSWER)

NEW QUESTION: 160

□

Mule □□□□□□□□ □□ □□ API □□□□□□ □□ □□□ □ □□ HTTP □□□□ □□□ □□□□.

http://acme.com/apis/orders □ http://acme .com/a pis/customers.

HTT□□ □□ □□ □□ □□ □□□□ □□□? □ HTTP □□□□ □□□□ □ □□□ □ □□□ □□□□ □□□?

- A. /apis/?
- B. /apis/
- C. /apis/orders|□□
- D. /apis/*

Answer: B (LEAVE A REPLY)

NEW QUESTION: 161

□□□ □□□□□.

2. 2. ---

3. 3. □□□ □□□("Todd.Pal@mulesoft.com")

Answer: D (LEAVE A REPLY)

* □□□ □□ □□□ □□□□□ DataWeave □□□□□ □□□ import □□□□ □□□□ □□
□□□ □□□ □□□□ □□□. □□ □□ □□□ □□□□.

1) String □□□□ □□□ □□□ □□□□ □□□□.

dw::core::□□□ □□□□

2) String □□□□ □□□ □□ □□□ □□□□□:

□□□□ camelize, dw::core::Strings□□ □□□□ □□

3) String □□□□ □□ □□□ □□□□□:

dw::core::Strings□□ * □□□□

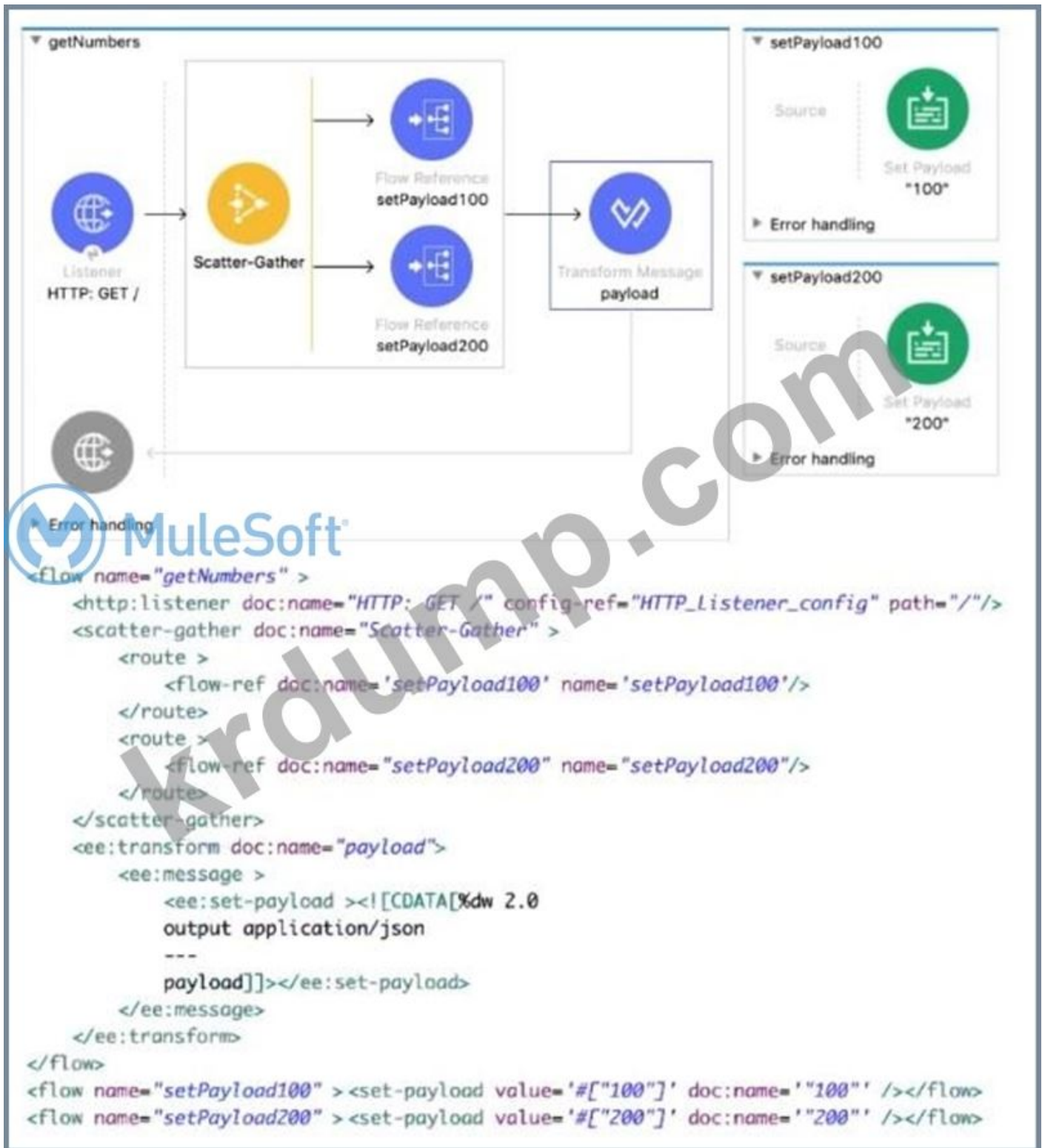
□□□ □□□□ □□□ DataWeave □□□□□□ □□ □□□ □□□□ □ □□□ □□□ □□
□ □□□. □□□□ □□□ □□ □□□ □□□□ □□□ * from□ □□□□ □□ □□□□ □□
□□□ □□□□ □□ □□□□□□ □□□ □□□ □ □□□ □□□□ □□□.

* □□□ □□□□□□ □□ WebStore.dwl□ □□□□ □□□ □□□□□.

□□□ □□□ □□□ □□□□.

NEW QUESTION: 163

□□□□ □□□□□.



□□□□ □□ □□□ □□ □□, □□ □ □□□□ □□ □□□ □□ □□□□□.

□□ □□□ □□□ □□□ □ Logger □□ □□□ □□□ □□□□ □□□□□□?

A)

```
[
  {
    "attributes": ...,
    "payload": "100"
  },
  {
    "attributes": ...,
    "payload": "200"
  }
]
```

B)

```
{
  "0": "100",
  "1": "200"
}
```

C)

```
["100", "200"]
```

D)

```
[
  "0": {
    "attributes": ...,
    "payload": "100"
  },
  "1": {
    "attributes": ...,
    "payload": "200"
  }
]
```

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 164

`http://localhost:8081?dept=sales`. dept
DataWeave `dept`?

- A. `dept`
- B. `vars.dept`
- C. `message.queryParams.dept`
- D. `queryParams.dept`

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 165

`putShipping`.



Mule `queryParams.dept`. HTTP Listener `queryParams.dept`.

HTTP Listener `queryParams.dept`?

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

Answer: A ([LEAVE A REPLY](#))

Input Output

Q type filter text

▼ Mule Message

▼ Payload

▼ Object : *Object*

▶ 0 : *Object*

▼ 1 : *Object*

▶ payload : *Array<Object>*

▶ attributes : *Object*

▼ 2 : *Object*

▼ payload : *Array<Object>*

airlineName : *String?*

availableSeats : *Number?*

departureDate : *String?*

destination : *String?*

flightCode : *String?*

origination : *String?*

planeType : *String?*

price : *Number?*

▶ attributes : *Object*

▼ Attributes

Void : *Void*

▼ Variables

▼ code

String : *String*



MuleSoft

Krdump.com

A.

B. API

C. API

D. API

Answer: ([SHOW ANSWER](#))

API API Exchange Design Center .

NEW QUESTION: 169

.



Message Flow Global Elements Configuration XML

Set to XML x Problems Console

There are no errors.

General

Metadata

Notes

Display Name: Set to XML

Settings

```
<ns2:listAllFlightsResponse xmlns:ns2="http://soap.training.mulesoft.com/">
<return>
```

Encoding:

MIME Type:


```

<flow name="getFruits">
  <http:listener doc:name="HTTP: GET /" config-ref="HTTP_listener_config" path="/" />
  <scatter-gather doc:name="Scatter-Gather">
    <route> <flow-ref doc:name="banana" name="banana"/> </route>
    <route> <flow-ref doc:name="apple" name="apple"/> </route>
  </scatter-gather>
  <sub-flow name="apple"><set-payload value="Apple" doc:name="Apple"/></sub-flow>
  <sub-flow name="banana"><set-payload value="Banana" doc:name="Banana"/></sub-flow>

```

http://localhost:8081/ . . . ?

A)

```
['Banana', 'Apple']
```

B)

```
{
  "0": "Banana",
  "1": "Apple"
}
```

C)

```
{
  "attributes": ...,
  "payload": ['Banana', 'Apple']
}
```

D)

```
{
  "0": {
    "attributes": ...,
    "payload": "Banana"
  },
  "1": {
    "attributes": ...,
    "payload": "Apple"
  }
}
```

A. C

B. D

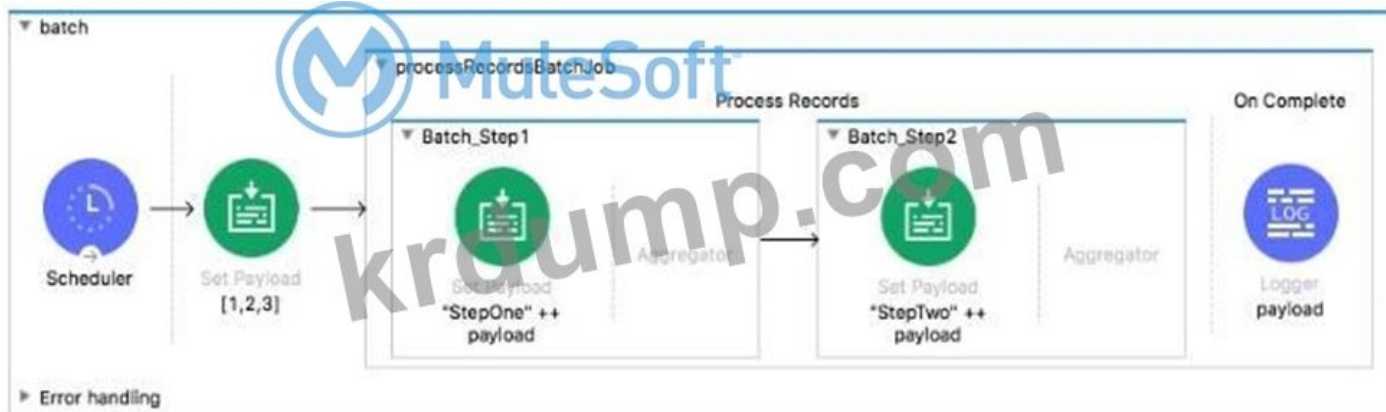
C. A

D. B

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 171

.



On Complete ?

- A. : [StepTwo1, StepTwo2, StepTwo3]
- B.
- C. : [StepTwoStepOne1, stepTwoStepOne2, StepTwoStepOne3]
- D. : [1,2,3]

Answer: ([SHOW ANSWER](#))

MCD-Level-1 DumpTop MCD-Level-1
! DumpTop **MCD-Level-1** , DumpTop MCD-Level-1
 . DumpTop
MCD-Level-1 . <https://www.dumptop.com/MuleSoft/MCD-Level-1-dump.html>
(235 Q&As Dumps, **30%OFF Special Discount: KrDump**)