MCD-LEVEL1-DELTA Q&As

MuleSoft Certified Developer - Level 1 (Mule 4) DELTA

Pass Mulesoft MCD-LEVEL1-DELTA Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/mcd-level1-delta.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft Official Exam Center

QUESTION 1

Refer to the exhibits.


```
<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"' />
```

A web client sends a POST request with the payload {"oid": "1000", "itemid": "AC200", "qty": "4" } to the Mule application. The File Write operation throws a FILE:CONNECTIVITY error. What response message is returned to the web client?

A. `\\'FILE:CONNECnvnY\\'

B. "ORDER:NOT_CREATED"

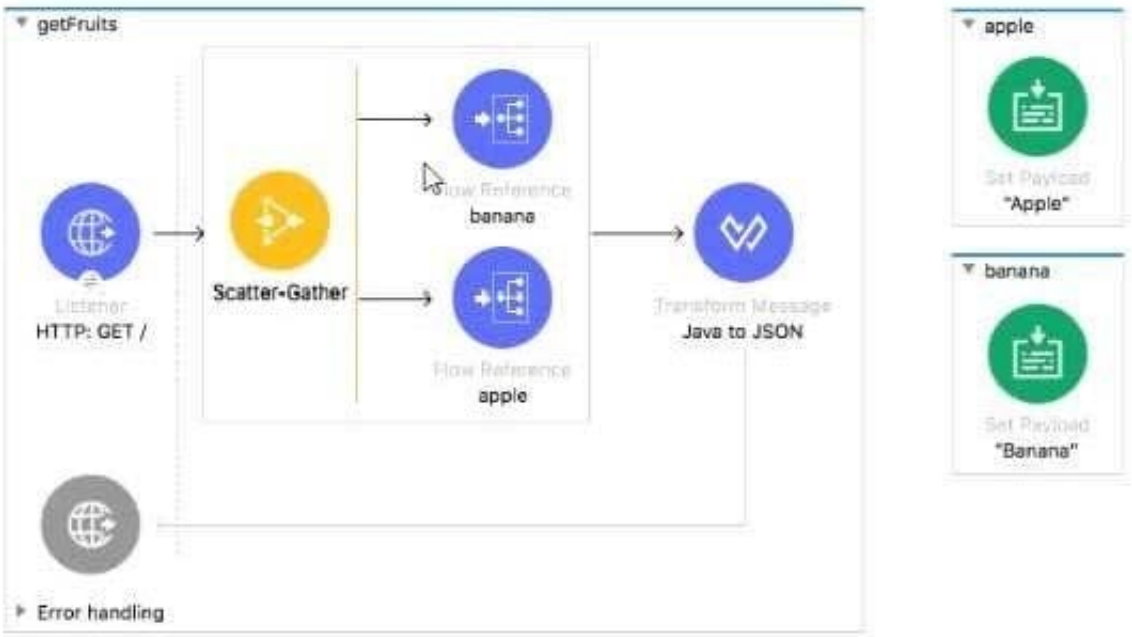
C. "OTHER ERROR"

D. "File written"

Correct Answer: A

QUESTION 2

Refer to the exhibits.




```
<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>
```

A web client submits a request to http://localhQst:8081. What is the structure of the payload at the end of the flow?

A. [\\'Banana\\', \\'Apple\\']

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

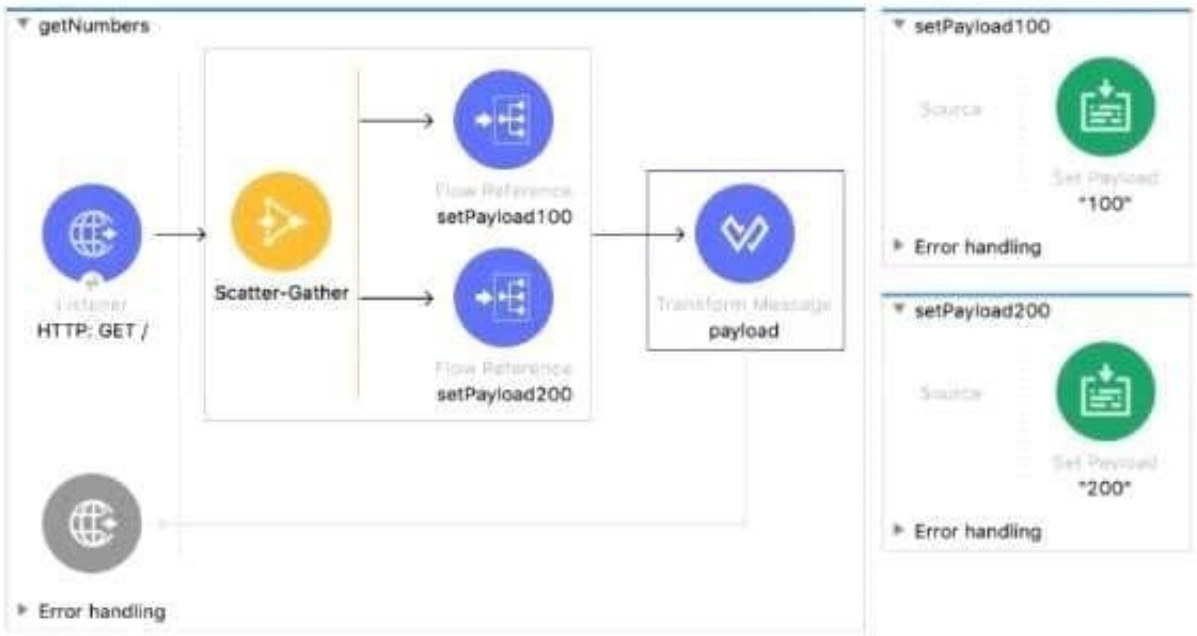
C. { "attributes": ..., "payload": [\\'Banana\\', \\'Apple\\'] }

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

Correct Answer: A

QUESTION 3

Refer to the exhibits.




```
<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>
```

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

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. Option A

B. Option B

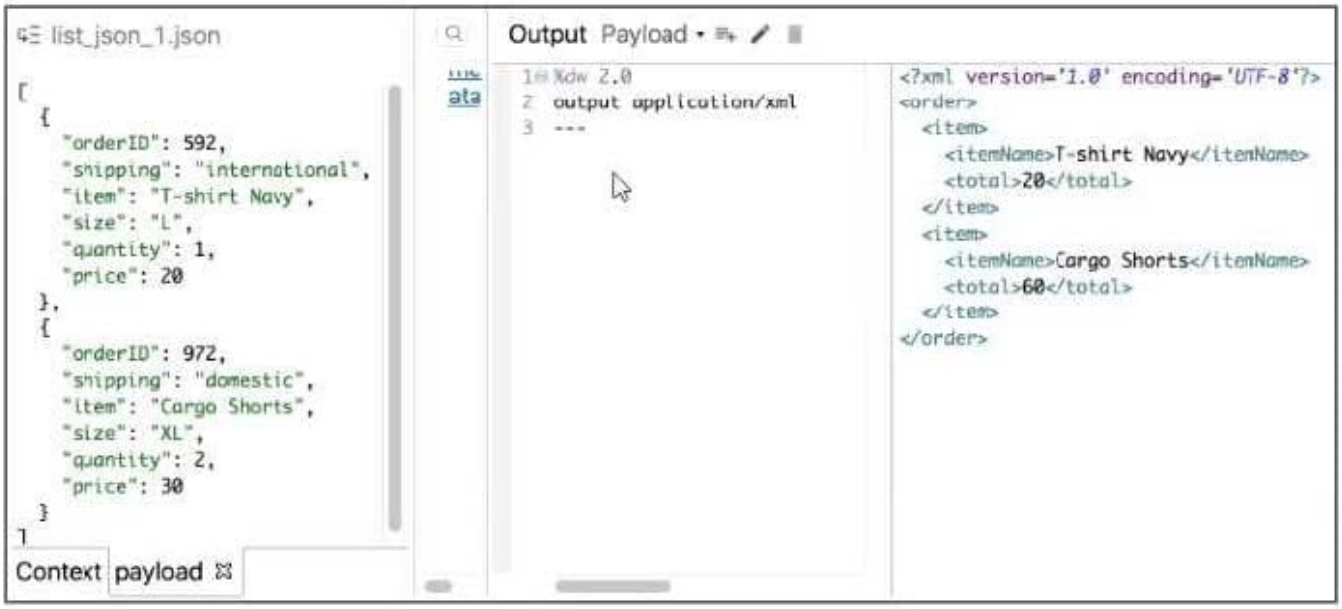
C. Option C

D. Option D

Correct Answer: C

QUESTION 4

Refer to the exhibit.



What Database expression transforms the input to the output?

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. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

QUESTION 5

What DataWeave 2.0 type can be used as input to a map operation?

A. Object

B. Array

C. String

D. Map

Correct Answer: B

Latest MCD-LEVEL1-DELTA Dumps | MCD-LEVEL1-DELTA PDF Dumps | MCD-LEVEL1-DELTA Practice Test