MCD-LEVEL1 Q&As

MuleSoft Certified Developer - Level 1 (Mule 4)

Pass Mulesoft MCD-LEVEL1 Exam with 100% Guarantee

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

https://www.leads4pass.com/mcd-level1.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

What of the below is not a feature of API Notebooks?

A. API documentation

B. Creates a client for an API

C. Creates a mock service for an API

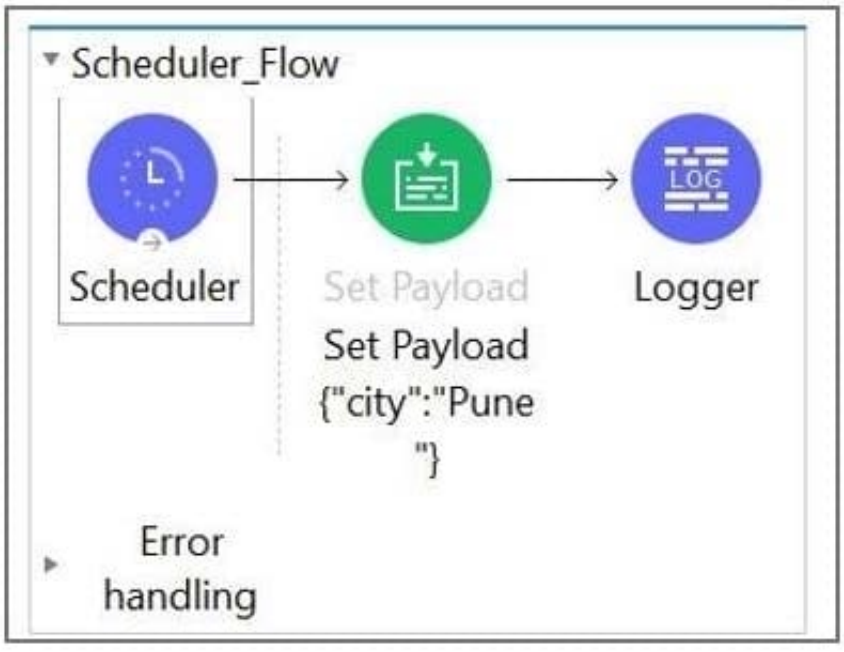
D. Perform authenticated live calls on a real server

Correct Answer: C

Correct answer is Creates a mock service for an API API Notebook is an open source, shareable web application for API documentation, interactive API tutorial and example generatation, and a client for your API endpoints. Using API Notebook, you can make requests and quickly transform the responses into readable format. However it cannot be used to mock service for an API. MuleSoft Doc Ref : https://docs.mulesoft.com/api-manager/1.x/api-notebook-concept

QUESTION 2

Refer to exhibits.



What message should be added to Logger component so that logger prints "The city is Pune" (Double quote should not be part of logged message)?

A. #["The city is" ++ payload.City]

B. The city is + #[payload.City]

C. The city is #[payload.City]

D. #[The city is ${payload.City}

Correct Answer: C

Correct answer is The city is #[payload.City]

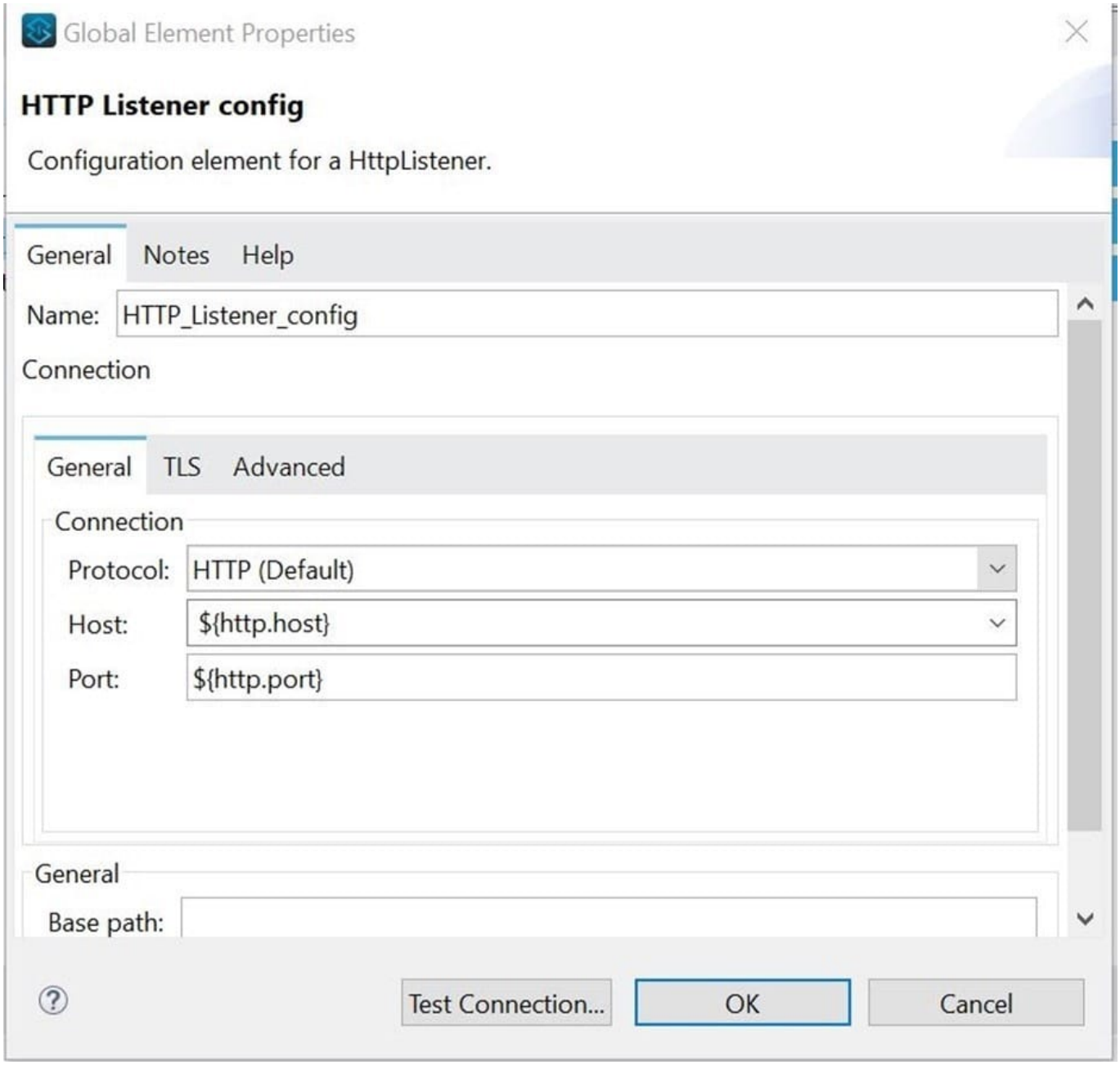
Answer can get confused with the option #["The city is" ++ payload.City] But note that this option will not print the space between is and city name. This will print The city isPune

QUESTION 3

Refer to the below exhibit.

A Mule application configures a property placeholder file named config.yaml to set some property placeholders for an HTTP connector.

What is the valid properties placeholder file to set these values?



A. 1.http:

2.

host = "localhost"

3.

port = "8081"

B. 1.http:

2.

basepath: "api"

3.

host : "localhost"

4.

port : "8081"

(Correct)

C. 1. http.host = localhost

2. http.port = 8081

D. 1. {

2.

http:

3.

basePath: "api",

4.

port: "8081",

5.

host: " localhost"

Correct Answer: B

Correct answer is as below as it follows the correct syntax http:

basepath: "api"

host : "localhost"

port : "8081"

QUESTION 4

A Utlility.dwl is located in a Mule project at src/main/resources/modules. The Utility.dwl file defines a function named encryptString that encrypts a String What is the correct DataWeave to call the encryptString function in a Transform Message component?

A. 1. %dw 2.0

2.

output application/json

3.

import modules::Utility

4.

--

5.

Utility::encryptString("John Smith")

B. 1. %dw 2.0

2.

output application/json

3.

import modules::Utility

4.

--

5.

encryptString("John Smith")

C. 1. %dw 2.0

2.

output application/json

3.

import modules.Utility

4.

--

5.

encryptString("John Smith")

D. 1. %dw 2.0

2.

output application/json

3.

import modules.Utility

4.

--

5.

Utility.encryptString("John Smith")

Correct Answer: B

Correct answer is %dw 2.0 output application/json import modules::Utility

Utility::encryptString("John Smith") DataWeave 2.0 functions are packaged in modules. Before you begin, note that DataWeave 2.0 is for Mule 4 apps. For Mule 3 apps, refer to DataWeave Operators in the Mule 3.9 documentation. For other Mule versions, you can use the version selector for the Mule Runtime table of contents. Functions in the Core (dw::Core) module are imported automatically into your DataWeave scripts. To use other modules, you need to import the module or functions you want to use by adding the import directive to the head of your DataWeave script, for example: import dw::core::Strings import camelize, capitalize from dw::core::Strings import * from dw::core::Strings The way you import a module impacts the way you need to call its functions from a DataWeave script. If the directive does not list specific functions to import or use * from to import all functions from a function module, you need to specify the module when you call the function from your script. For example, this import directive does not identify any functions to import from the String module, so it calls the pluralize function like this: Strings::pluralize("box"). Transform %dw 2.0 import dw::core::Strings output application/json

{ \\'plural\\': Strings::pluralize("box") }

QUESTION 5

According to MuleSoft. what is the first step to create a Modern API?

A. Gather a list of requirements to secure the API

B. Create an API specification and get feedback from stakeholders

C. Performance tune and optimize the backend systems and network

D. Create a prototype of the API implementation

Correct Answer: B

First step in creating Modern API is to create an API specification and get feedback from stakeholders so that any future issues can be identified at early stage thereby reducing overall delivery time Reference: https://developer.mulesoft.com/tutorials-and-howtos/quick-start/designing-your-first-api