350-901 Q&As

Developing Applications Using Cisco Core Platforms and APIs (DEVCOR)

Pass Cisco 350-901 Exam with 100% Guarantee

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

https://www.passapply.com/350-901.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

DRAG DROP

Refer to the exhibit. The Python script is supposed to make an API call to Cisco DNA Center querying a wireless profile for the "ChicagoCampus" and then parsing out its enable FlexConnect value. Drag and drop the parts of the Python code from the left onto the item numbers on the right that match the missing sections in the exhibit.

GET /dna/intent/api/v1/wireless/profile Get Wireless Profile

Gets either one or all the wireless network profiles if no name is provided for network-profile.

Parameters

Name	Description
profileName string *(query)*	*Default value:*

Responses

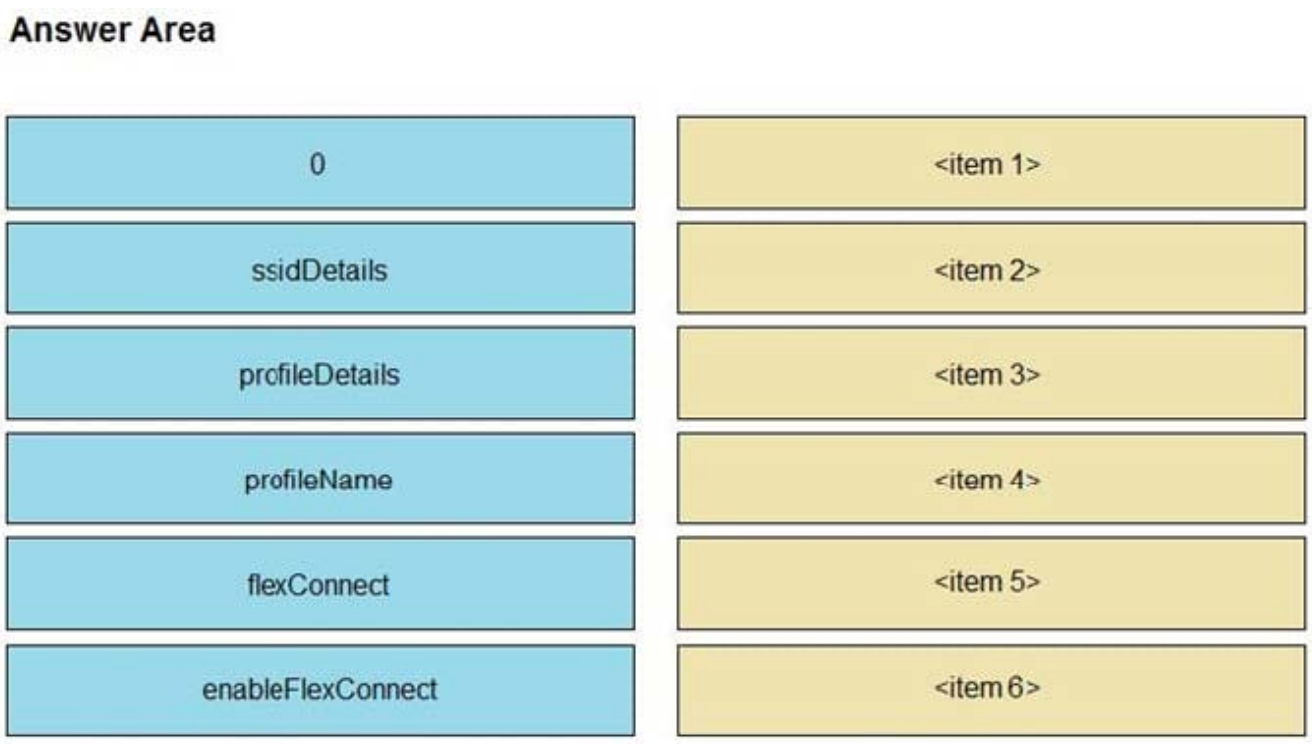
Code	Description
200	*The request was successful. The result is contained in the response body.*

Example Value Model

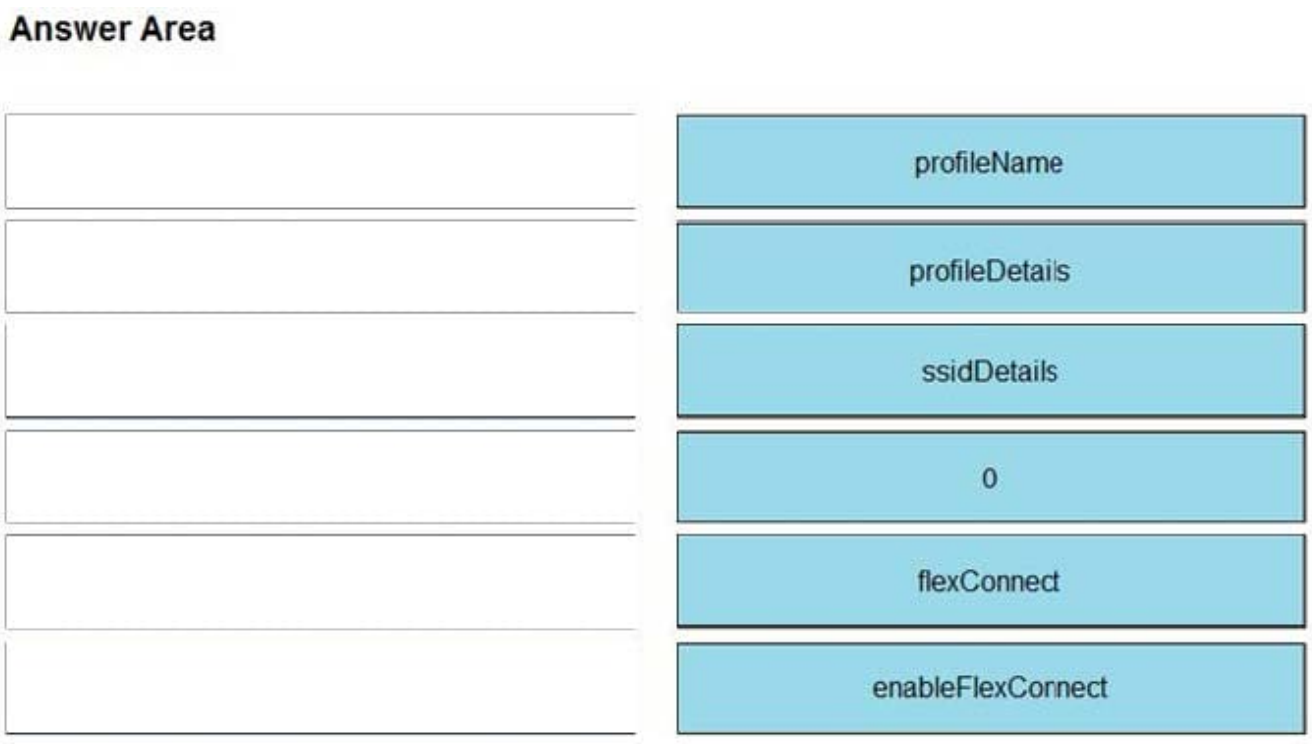
```
[
  {
    "profileDetails": {
      "name": "string",
      "sites": [
        "string"
      ],
      "ssidDetails": [
        {
          "name": "string",
          "type": "Guest",
          "enabledFabric": true,
          "flexConnect": {
            "enableFlexConnect": true,
            "localToVlan": 0
            },
            "InterfaceName": "string"
          }
        ]
      }
    }
  }
]
```

```
import requests
import json

def get_dnac_wireless_profiles():
    try:
      url = "https://sandboxdnac2.cisco.com/dna/intent/api/v1" \
      + "/wireless/profile?<item1>=ChicagoCampus|"

      print(token)
      payload = {}
      headers = {
      'x-auth-token': token
      }

      response = requests.request("GET", url, headers=headers, data = payload)
      response.raise_for_status()
      return response.json()[0]['<item 2>'][' <item 3>'] \
                          [<item 4>]['<item 5>']["<item 6>"]
    except Exception as e:
      print(e)

def create_dnac_token():
    try:
      url = "https://sandboxdnac2.cisco.com/dna/system/api/v1/auth/token"

      payload = {}
      headers = {
      'Authorization': 'Basic ZGV2bmV0dXNlcjpDaXNjbzEyMyE= ',
      'Content-Type': 'application/json'
      }

      response = requests.request("POST", url, headers=headers, data = payload)
      response.raise_for_status()
      return response.json()["Token"]
    except Exception as e:
      print(e)

if __name__ == "__main__":
    token = create_dnac_token()
    print(get_dnac_wireless_profiles())
```

Select and Place:



Correct Answer:



QUESTION 2

Refer to the exhibit.

```
import requests
import getpass

device_list = ['192.168.243.1', '192.168.243.2']
port = "8080"

username = input("Enter Username  -->")
password = getpass.getpass(prompt="Enter Password:  ->")

for device in device_list:
    [                                                    ]

    headers = {'Content-Type': 'application/vnd.yang.data+json',\
               'Accept': 'application/vnd.yang.data+json'}

    response = requests.get( url, auth=(username, password),\
                             headers=headers, verify=False)

    print(f"Interfaces present on  {device}:")
    for interfaces in response.json():
        print(f"{interfaces}")
```

Cisco IOS XE switches are used across the entire network and the description that is filed for all interfaces must be configured. Which code snippet must be placed in the blank in the script to leverage RESTCONF to query all the devices in the device list for the interfaces that are present?

A.
```
url="http://"+device+":"+port+"/api/running/
     interfaces"
```

B.
```
url="http://"+device+":"+port"/api/running/
     interfaces/interface"
```

C.
```
url="http://"+device+":"+port"/api/running/
     interfaces/interface/name"
```

D.
```
url=http://f"{device_list}+{port}/api/running/
    interfaces"
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: A

QUESTION 3

Refer to the exhibit.

```
def init_tracer(service):
    logging.getLogger('').handlers = []
    logging.basicConfig(format='%(message)s', level=logging.DEBUG)
    config = Config(
        config={'sampler': {'type': 'const', 'param': 1,},'logging': True,},
        service_name=service,)
    return config.initialize_tracer()

base_url = 'https://sandboxdnac.cisco.com/'

        try:
            dnac = DNACenterAPI(username='devnetuser', password='Cisco123!',
                                base_url=base_url, version='1.3.3',
                                verify=False)
            print('auth passed')
        except Exception as e:
            print('failed')
    try:
        devices = dnac.devices.get_device_list()
        devices = devices['response']

        devices = [device['hostname'] for device in devices]
        print(devices)
    except Exception as e:
```

An application is created to serve the needs of an enterprise. Slow performance now impacts certain API calls, and the application design lacks observability. Which two commands improve observability and provide an output that is similar to the sample output? (Choose two.)

A.
```
dnac-tracer3 = init_tracer('dnac-tracer3')
```

B.
```
with dnac-tracer3.start_span('dnac-api-calls') as span:
```

C.
```
with tracer.start_span('dnac-api-calls') as span:
```

D.
```
dnac-tracer3.start_span('dnac-api-calls') as span:
```

E.
```
tracer = init_tracer('dnac-tracer3')
```

A. Option A

B. Option B

C. Option C

D. Option D

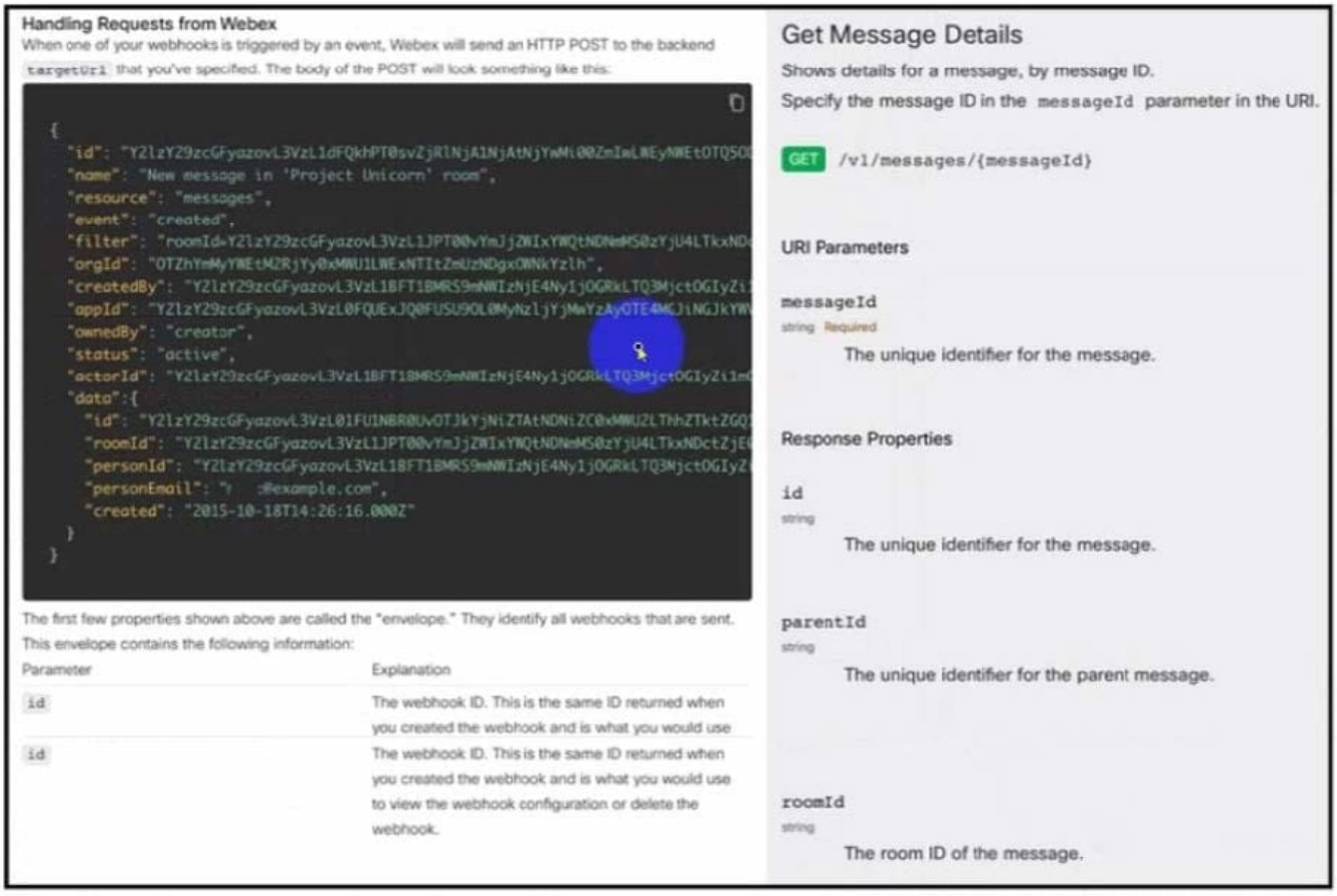
E. Option E

Correct Answer: CE

QUESTION 4

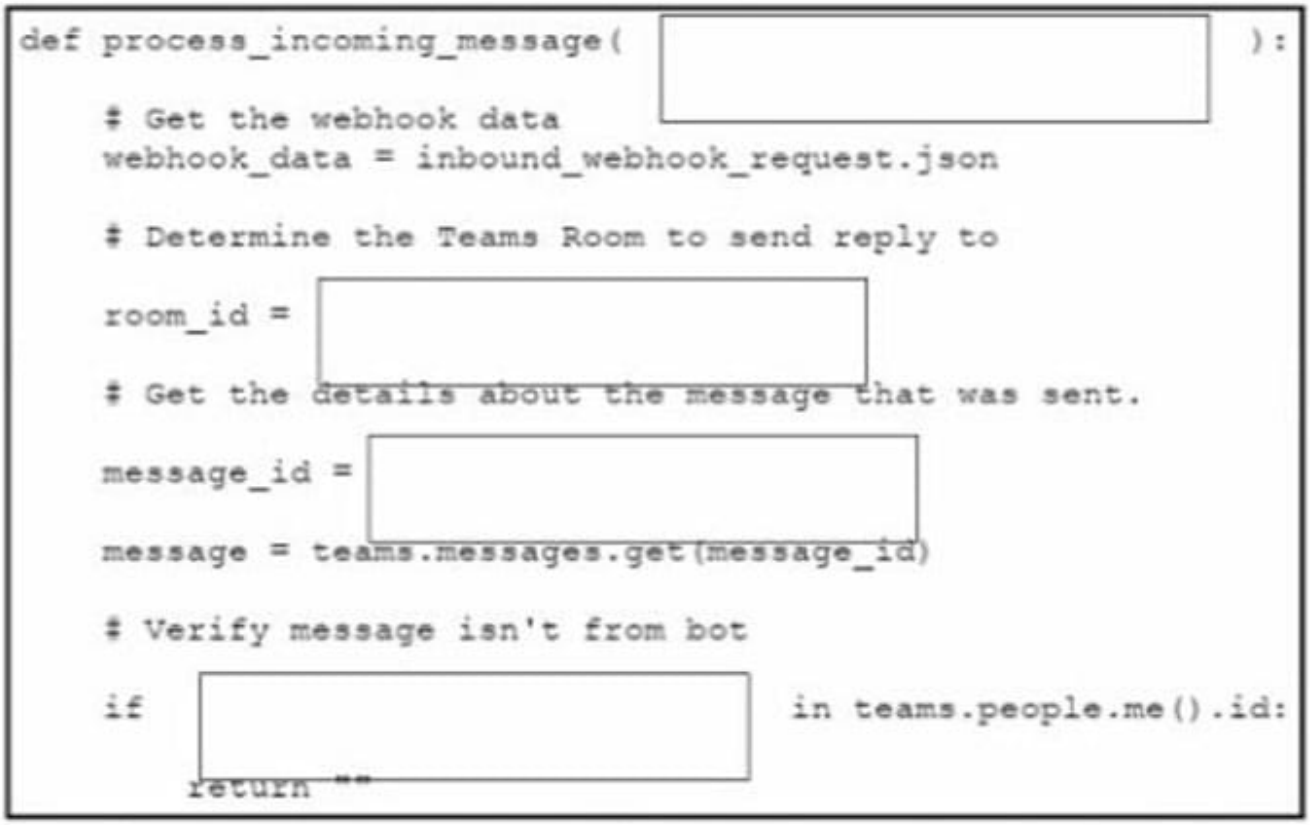
DRAG DROP

Refer to the exhibit.



Drag and drop the code from the bottom onto the box where the code is missing to complete the API request. An engineer is using this API request to implement Chat-Ops to generate notifications m a Webex space by using webhooks. Not all options are used.

Select and Place:



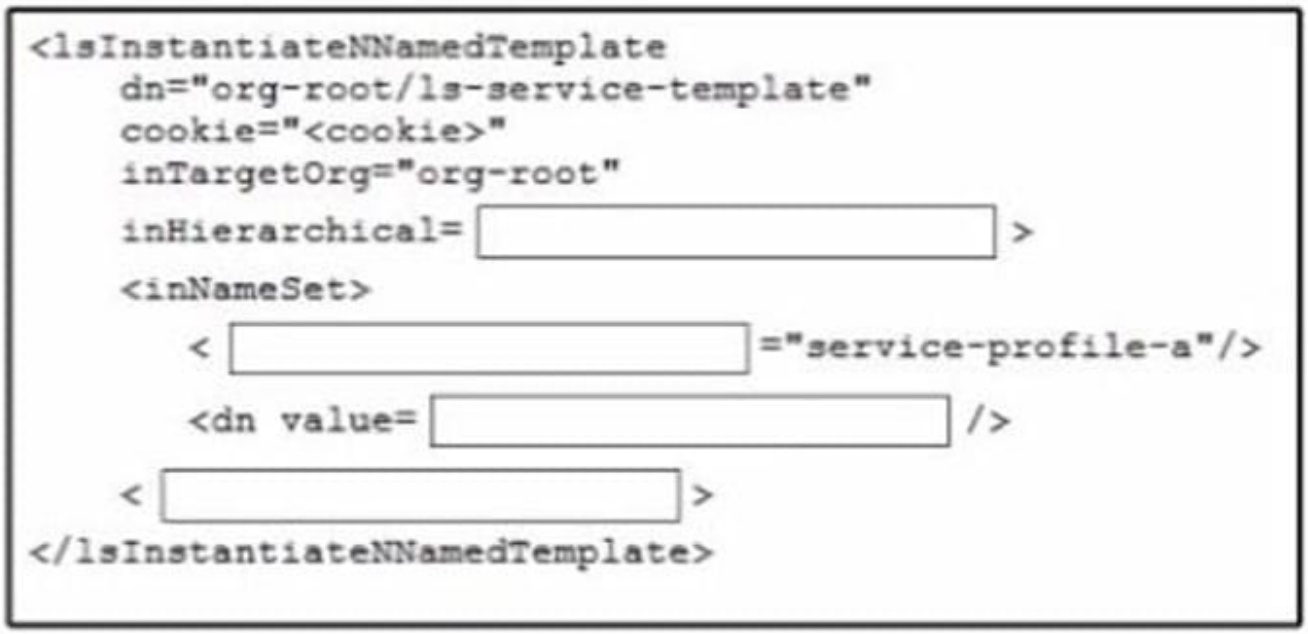
Correct Answer:

QUESTION 5

DRAG DROP

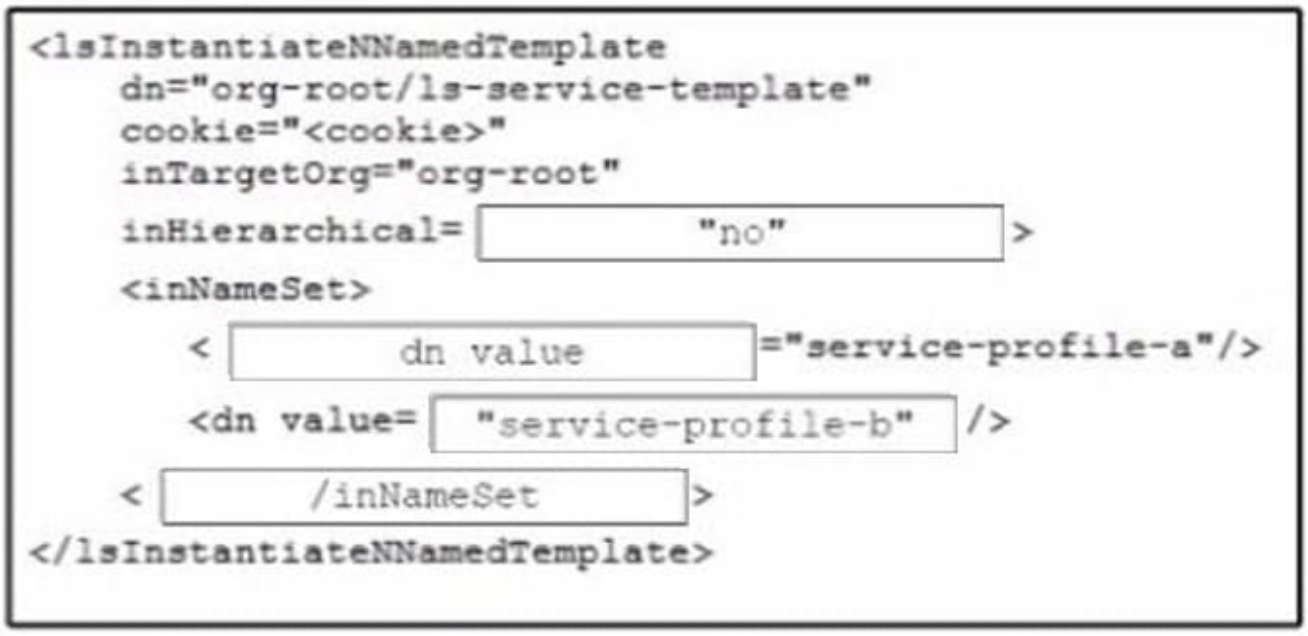
Drag and drop the code from the bottom onto the box where the code is missing to construct a UCS XML API request to generate two service profiles from the template org- root/is-service-template. Not at options are used.

Select and Place:



"service-profile-b"
"yes"
dn value
/outNameSet
/inNameSet
add profile
"no"

Correct Answer:



[350-901 VCE Dumps](#) [350-901 Exam Questions](#) [350-901 Braindumps](#)