300-635 Q&As

Automating and Programming Cisco Data Center Solutions (DCAUTO)

QUESTION 1

What is the default data encoding for the response output of the ACI APIC API inspector?

A. CSV

B. JSON

C. XML

D. YAML

Correct Answer: A

QUESTION 2

When should the API Inspector be used?

A. to send an API request to the APIC

B. to learn or identify the sequence of API calls for a specific operation in the APIC GUI

C. to verify the XML structure of an object based on a specific operation in the APIC GUI

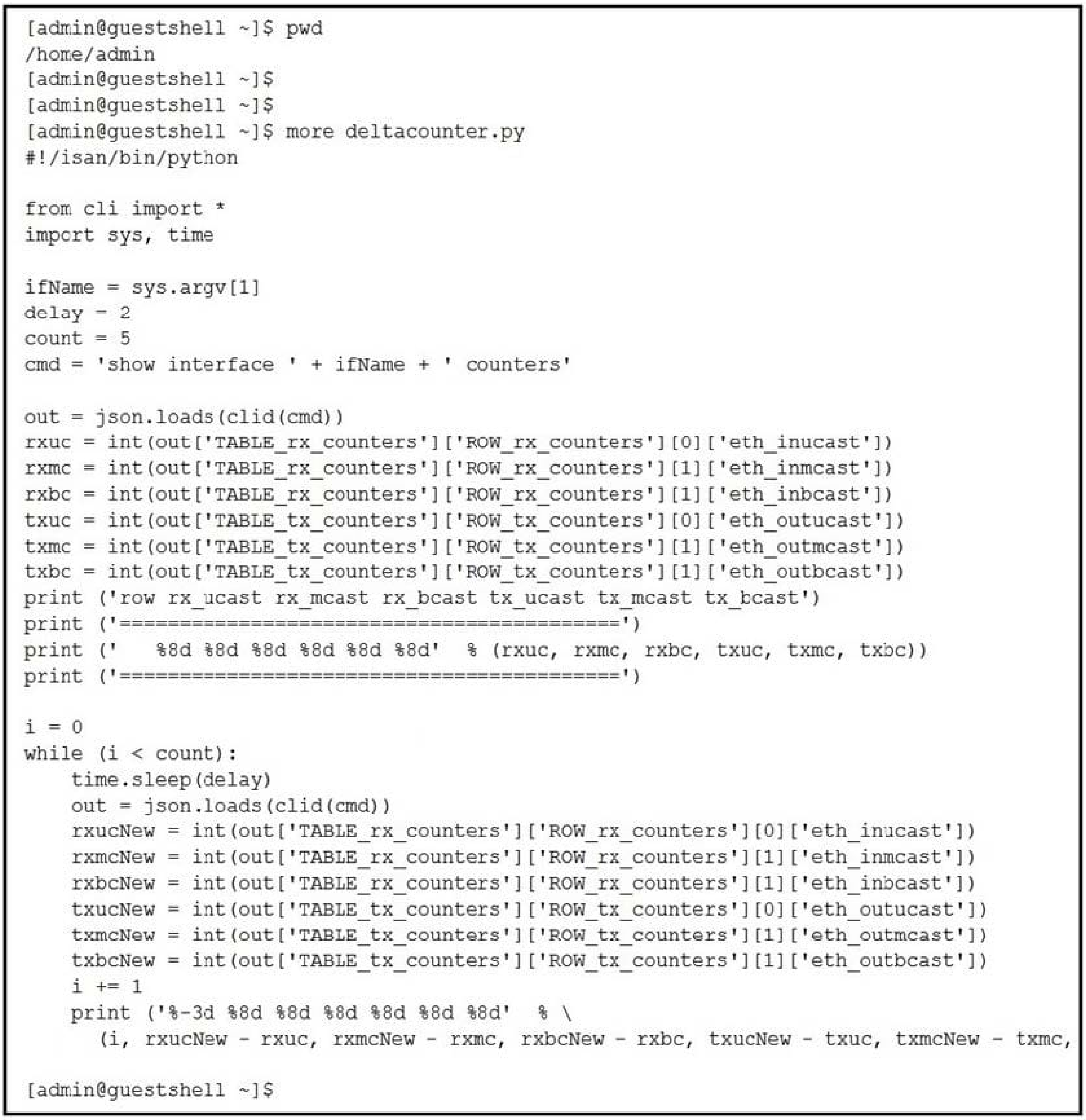
D. to launch an Ansible playbook

Correct Answer: B

Devnet training

QUESTION 3

Refer to the exhibit.



```
[admin@guestshell ~]$ pwd
/home/admin
[admin@guestshell ~]$
[admin@guestshell ~]$
[admin@guestshell ~]$ more deltacounter.py
#!/isan/bin/python

from cli import *
import sys, time

ifName = sys.argv[1]
delay - 2
count = 5
cmd = 'show interface ' + ifName + ' counters'

out = json.loads(clid(cmd))
rxuc = int(out['TABLE_rx_counters']['ROW_rx_counters'][0]['eth_inucast'])
rxmc = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inmcast'])
rxbc = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inbcast'])
txuc = int(out['TABLE_tx_counters']['ROW_tx_counters'][0]['eth_outucast'])
txmc = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outmcast'])
txbc = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outbcast'])
print ('row rx_ucast rx_mcast rx_bcast tx_ucast tx_mcast tx_bcast')
print ('==========================================')
print ('    %8d %8d %8d %8d %8d %8d'  % (rxuc, rxmc, rxbc, txuc, txmc, txbc))
print ('==========================================')

i = 0
while (i < count):
    time.sleep(delay)
    out = json.loads(clid(cmd))
    rxucNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][0]['eth_inucast'])
    rxmcNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inmcast'])
    rxbcNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inbcast'])
    txucNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][0]['eth_outucast'])
    txmcNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outmcast'])
    txbcNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outbcast'])
    i += 1
    print ('%-3d %8d %8d %8d %8d %8d %8d'  % \
      (i, rxucNew - rxuc, rxmcNew - rxmc, rxbcNew - rxbc, txucNew - txuc, txmcNew - txmc,

[admin@guestshell ~]$
```

The script is called deltacounters.py and it is currently inside a Guest Shell container running inside a Cisco NX-OS switch. Which Cisco NX-OS command results in a successful execution of this script?

A. python /home/admin/bootflash:deltacounters.py ethemet1/1

B. show python bootflash:deltacounters.py ethernet1/1

C. guestshell run python /home/admin/deltacounter.py ethernet1/1

D. guestshell execute python /home/admin/deltacounter.py ethernet1/1

Correct Answer: C

Reference: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b_166_programmability_cg/guest_shell.html

QUESTION 4

During the process of starting a Python network telemetry collector, which command starts the Cisco bigmuddy-network-telemetry-collector from GitHub?

A. model driven telemetry

B. telemetry_receiver.py --ip-address --port

C. telemetry_receiver.py --destination --url

D. streaming telemetry

Correct Answer: B

Reference:

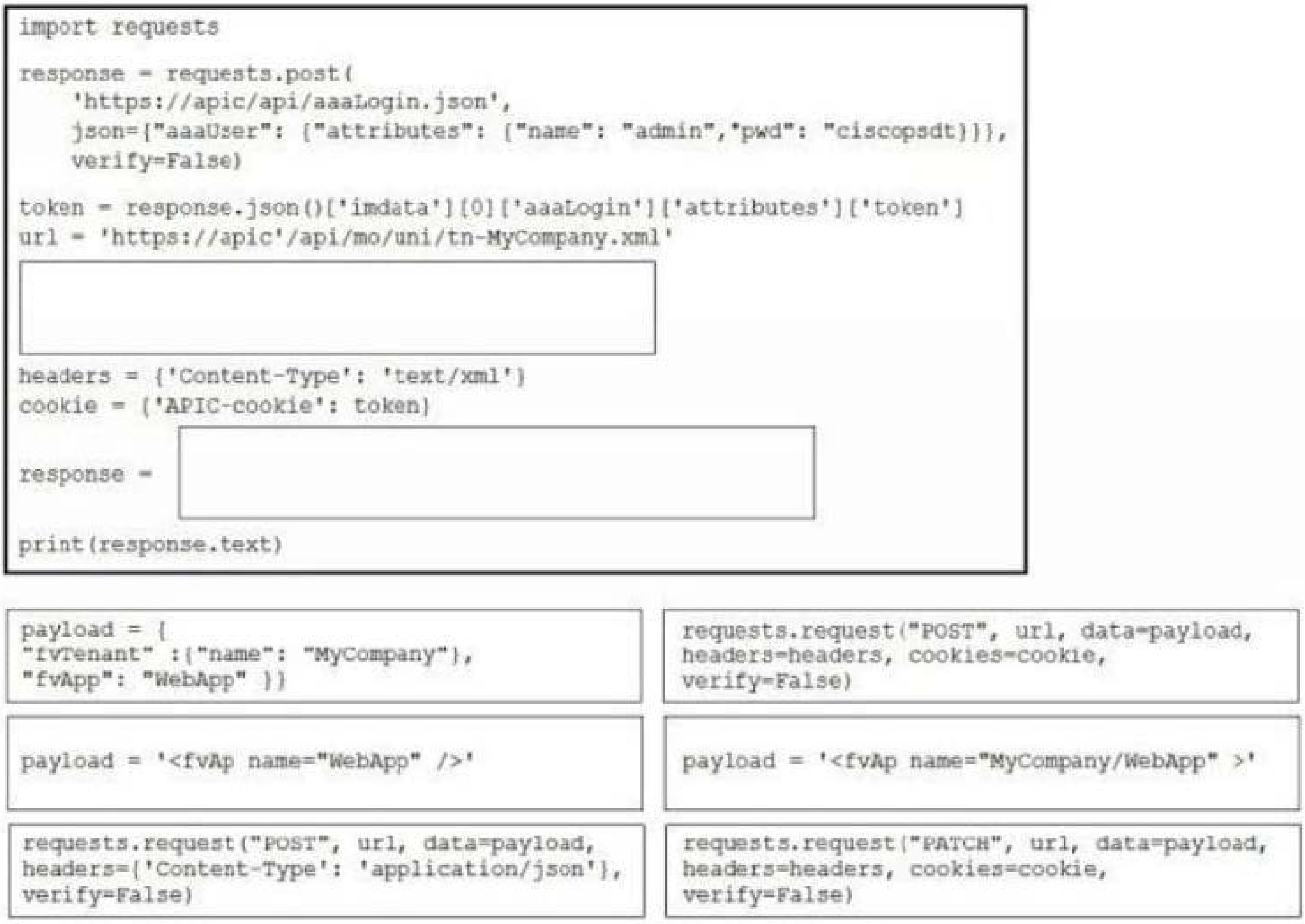
https://developer.cisco.com/codeexchange/github/repo/cisco/bigmuddy-network-telemetry-collector/#:~:text=8,bigmuddy%2Dnetwork%2Dtelemetry%2Dcollector,use%20of%20the%20resulting%20data.

QUESTION 5

DRAG DROP

Drag and drop the correct code snippets into the Python code to create a new application profile "WebApp" using the ACI REST API. Not all options are used.

Select and Place:



```
import requests

response = requests.post(
    'https://apic/api/aaaLogin.json',
    json={"aaaUser": {"attributes": {"name": "admin","pwd": "ciscopsdt}}},
    verify=False)

token = response.json()['imdata'][0]['aaaLogin']['attributes']['token']
url = 'https://apic'/api/mo/uni/tn-MyCompany.xml'

headers = {'Content-Type': 'text/xml'}
cookie = {'APIC-cookie': token}

response = 

print(response.text)
```

```
payload = {
"fvTenant" :{"name": "MyCompany"},
"fvApp": "WebApp" }}
```

```
requests.request("POST", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

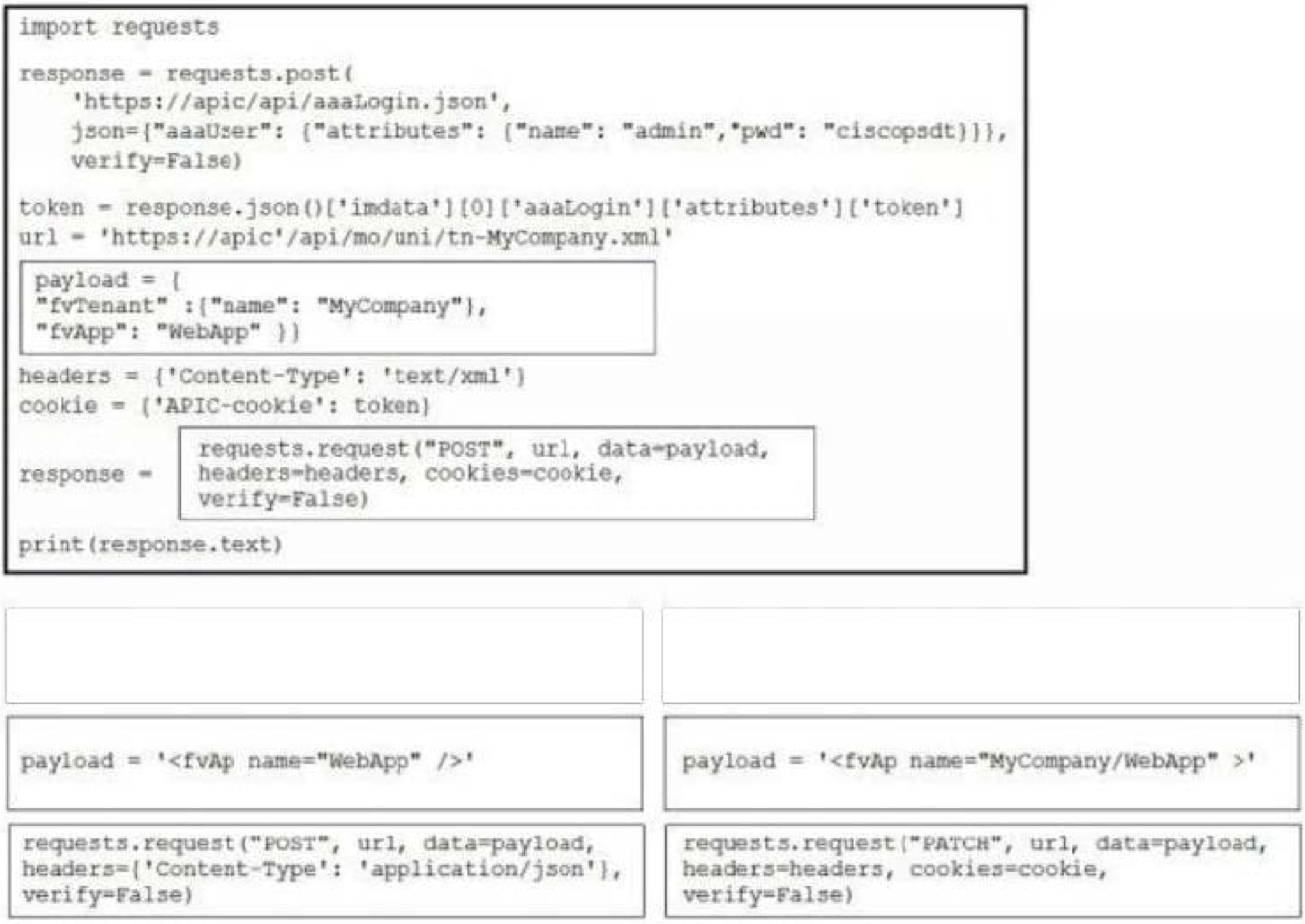
```
payload = '<fvAp name="WebApp" />'
```

```
payload = '<fvAp name="MyCompany/WebApp" >'
```

```
requests.request("POST", url, data=payload,
headers={'Content-Type': 'application/json'},
verify=False)
```

```
requests.request("PATCH", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

Correct Answer:



```
import requests

response = requests.post(
    'https://apic/api/aaaLogin.json',
    json={"aaaUser": {"attributes": {"name": "admin","pwd": "ciscopsdt}}},
    verify=False)

token = response.json()['imdata'][0]['aaaLogin']['attributes']['token']
url = 'https://apic'/api/mo/uni/tn-MyCompany.xml'
```

```
payload = {
"fvTenant" :{"name": "MyCompany"},
"fvApp": "WebApp" }}
```

```
headers = {'Content-Type': 'text/xml'}
cookie = {'APIC-cookie': token}
```

```
response =
```

```
requests.request("POST", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

```
print(response.text)
```

```
payload = '<fvAp name="WebApp" />'
```

```
payload = '<fvAp name="MyCompany/WebApp" >'
```

```
requests.request("POST", url, data=payload,
headers={'Content-Type': 'application/json'},
verify=False)
```

```
requests.request("PATCH", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

Latest 300-635 Dumps   300-635 Practice Test   300-635 Exam Questions