

300-635^{Q&As}

Automating and Programming Cisco Data Center Solutions (DCAUTO)

Pass Cisco 300-635 Exam with 100% Guarantee

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

<https://www.leads4pass.com/300-635.html>

100% Passing Guarantee
100% Money Back Assurance

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

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



QUESTION 1

DRAG DROP

Drag and drop the items to complete the request to retrieve the current firmware of Cisco UCS devices from the Cisco Intersight API. Not all items are used.

Select and Place:

<input type="text"/>	"https://intersight.com/api/v1
<input type="text"/>	/ <input type="text"/> "

- | | |
|-----------------|------------------|
| asset | RunningFirmwares |
| UpgradeStatuses | POST |
| GET | firmware |

Correct Answer:

GET	"https://intersight.com/api/v1
firmware	/ RunningFirmwares "

- | | |
|-----------------|------|
| asset | |
| UpgradeStatuses | POST |
| | |

Reference: <https://developer.cisco.com/codeexchange/github/repo/CiscoUcs/intersight-python/>

QUESTION 2

Using the NX-API CLI JSON-RPC interface, which two Python data structure and requests call create an SVI? (Choose two.)

- A.

```
requests.post(url, data=json.dumps(payload), headers={'content-type': 'application/json-rpc'}, auth=(username, password))
```
- B.

```
requests.post(url, data=json.dumps(payload), headers={'content-type': 'application/json'}, auth=(username, password))
```
- C.

```
payload = {
    "jsonrpc": "2.0", "method": "cli_conf",
    "params": {
        "command": "conf t; interface vlan " + id,
        "version": 1},
    "id": 1
}
```
- D.

```
payload = [
    {
        "jsonrpc": "2.0", "method": "cli",
        "params": {"cmd": "conf t", "version": 1},
        "id": 1
    },
    {
        "jsonrpc": "2.0", "method": "cli",
        "params": {"cmd": "interface vlan " + id, "version": 1},
        "id": 2
    }
]
```
- E.

```
payload = {
    "jsonrpc": "2.0", "method": "cli_conf",
    "params": {"cmd": "interface vlan " + id, "version": 1},
    "id": 1
}
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Correct Answer: AD

Sandbox

QUESTION 3

A set of automation scripts work with no issue from a local machine, but an experiment needs to take place with a new package found online. How is this new package isolated from the main code base?

- A. Add the new package to your requirements.txt file.
- B. Create a new virtual machine and perform a pip install of the new package.
- C. Perform a pip install of the new package when logged into your local machine as root.
- D. Create a new virtual environment and perform a pip install of the new package.

Correct Answer: D

Devnet Training

QUESTION 4

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": "ciscopdt"}},
    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)

QUESTION 5

Which authentication method is used when the REST API of the Cisco UCS Director is accessed?

- A. Bearer ((Bearer Token))
- B. HTTP Basic Auth
- C. RestAuth: ((User\'s Auth Token))
- D. X-Cloupia-Request-Key: ((User\'s Auth Token))

Correct Answer: D

Reference : https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-director/rest-api-cookbook/6-6/cisco-ucs-director-REST-API-cookbook-66/cisco-ucs-director-REST-API-cookbook-66_chapter_010.html