Free Questions for 350-901 by certsinside

Shared by Jordan on 05-09-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

{'lat': 37.4180951010362, 'lng': -122.098531723022, 'address': '', 'serial': 'Q2HP-F5K5-F98Q',

'mac': '88:15:44:ea:f5:bf', 'lanIp': '10.10.10.15',

'model': 'MS220-8P', 'switchProfileId': None, 'firmware': 'switch-11-31', 'floorPlanId': None}

Refer to the exhibit. A developer needs to find the geographical coordinates of a device on the network L_397561557481105433 using a Python script to query the Meraki API. After running response = requests.get() against the Meraki API, the value of response.text is shown in the exhibit.

What Python code is needed to retrieve the longitude and latitude coordinates of the device?

Options:

A) latitude = response.text['lat']
longitude = response.text['lng']

B) latitude = response.json()['lat']
longitude = response.json()['lng']

C) latitude = response.json()[0]
longitude = response.json()[1]

D) latitude = response.text[0]
longitude = response.text[1]

Answer:

A

Question 2

Question Type: **MultipleChoice**

Meraki Dashboard API Response

Response Status Code 200

Response Link Header

0000-0000-0000>; rel-first,

&startingAfter-Q2EK-3UBE-RRUY>; rel-next,

zzzz&perPage-3>; rel-last Response Body

{

'serial': 'Q2CV-V49B -RCMZ',

'mac': ''0c:8d:db:95:aa:39',

'networkid': 'L-566327653141846927',

'model11 : 'MV71',

'address': '430 E Cactus Ave .\nLas Vegas, NV 89183',

'lat': 36.00017,

'lng': -115.15302,

'notes': "',

'tagsn: '',,

'lanip: '192.168.0.25',

' configurationUpdatedAt': '2019-08-08T02:15:36Z', ' firmware11 : 'ca.rnera-3-3011

},

{

'ncune': 'Alex's MR84 - 1'1

'serial': 'Q2EK-2LYB-PCZP',

'mac': 'eO: 55:3d:10:56:8a', 'networkid': 'L 566327653141846927',

'model': 'MR84',

'address': ''11 ,

'lat': 39.9482993357826,

'lng': -82.9895675461739,

'notes': '',

'tags': ' ',

'lanip: null,

'configurationVpdatedAt': '2018-02-03Tll:02:37Z',

'firmware11 : 'Not running configured version''

},

{

'na.rne11 : 'Vegas Living Room MR84 11 ,

'serial': 'Q2EK-3UBE-RRUY',

'mac': 'e0:55:3d:10:5a:ca', 'networkid': 'L_566327653141846927' 1

'model': 'MR84',

'address': '430 E Cactus Ave.\nLas vegas, NV 89183', 'lat': 36.00015,

'lng': -115.15308,

'notes': ' ',

'tags': '11

1

'lanip: '192.168.0 .20',

'configurationVpdatedAt': '2018-09-29Tl2:23:21Z',

' firmware': 'Not running configured version'

Refer to the exhibit.

Image not found or type unknown

Which line of code must be added to this code snippet to allow an application to pull the next set of paginated items?

Options:

A) requests.get(url, links=['next']['url'])

B) requests.get(url, headers=links['next']['url'])

C) requests.get(res.links['next']['url'], headers=headers)

D) requests.get(res.headers.get('Link'')['next']['url'], headers=headers)

Answer:

C

Question 3

Question Type: **DragDrop**

Resource Tab:

add_mo

commit

logout

UcsHandle

ComputePooledSlot

ComputePool

LsRequirement

ted

isconnecting

```
le import UcsHandle

n handle
handle = UcsHandle("192.168.1.1", "admin", "password")

handle.login()

# Logout from the server
handle.logout()
```

Refer UcsHandle API Reference for detailed parameter sets to `UcsHandle`

This module contains the general information for ComputePooledSlot ManagedObject.

add_mo

class

ucsmsdk.mometa.compute.ComputePooledSlot.**ComputePooledSlot**(*parent_mo_or_dn*, [source]

commit

Bases: **ucsmsdk.ucsmo.ManagedObject**

logout

consts = *<ucsmsdk.mometa.compute.ComputePooledSlot.ComputePooledSlot-*

UcsHandle

oMeta object>

naming_props = *set([u'chassisId', u'slotId'])*

ComputePooledSlot

', 'sacl': 'sacl', 'slotId': 'slot_id', 'assigned': 'assigned', 'owner': 'owner', 'prevAssignedToDn': 'prev_assigned_to_dn', 'child-

ComputePool

oolable_dn', 'chassisId': 'chassis_id', 'rn': }

prop_meta = *{'dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at*

LsRequirement

coremeta.MoPropertyMeta object at 0x1233ad5a0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad4d0>, 'assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392b10>, 'assigned': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392bd0>, 'owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad2d0>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad1d0>, 'poolable_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad350>, 'chassis_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392ad0>, 'slot_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad550>, 'prev_assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad3d0>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad450>}

utePool.**ComputePool**(*parent_mo_or_dn*, [source]

add_mo

Bases: **ucsmsdk.ucsmo.ManagedObject**

commit

consts = *<ucsmsdk.mometa.compute.ComputePool.ComputePoolConsts*

logout

mo_meta = *<ucsmsdk.ucscoremeta.MoMeta object>*

UcsHandle

us', 'policyLevel': 'policy_level', 'assignment-Order': 'assignment_order', 'sacl': 'sacl', 'policyOwner': 'policy_owner',

ComputePooledSlot

, 'childAction': 'child_action', 'name': 'name', 'descr': 'descr', 'rn': 'rn', 'size': 'size'}

ComputePool

remeta.MoPropertyMeta object at

scoremeta.MoPropertyMeta object at

0x1230ed3d0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at

LsRequirement

k.ucscoremeta.MoPropertyMeta object at 0x1230f8d90>, 'int_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed050>, 'assignment_order': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e10>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e90>, 'name': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed0d0>, 'descr': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8f10>, 'policy_owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed1d0>, 'policy_level': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed150>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed250>, 'size': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed350>}

ent.**LsRequirement**(*parent_mo_or_dn,*
[source]

add_mo

Bases: **ucsmsdk.ucsmo.ManagedObject**

add_mo

```
from ucsmsdk.mometa.compute.ComputePool import ComputePool
from ucsmsdk.mometa.compute.ComputePooledSlot import ComputePooledSlot
```

commit

```
    "admin",
    "password"
```

logout

```
SERVER_POOL = <item 2>(
    parent_mo_or_dn="org-root/org-devnet",
```

UcsHandle

```
for blade in HANDLE.query_classid(
    "computeBlade",
```

ComputePooledSlot

```
    parent_mo_or_dn=SERVER_POOL,
    chassis_id=blade.chassis_id,
```

ComputePool

```
SP_TEMPLATE = <item 5>(
```

LsRequirement

```
HANDLE.<item 6>()

HANDLE.<item 7>()
```

LsRequirementConsts instance>
ct>

te': 'oper_state', 'qualifier': 'quali-
tion': 'restrict_migration', 'issues':
on_pool_dn', 'name': 'name',
dAction': 'child_action', 'as-

PropertyMeta object at 0x122cf-
rtyMeta object at 0x12e892790>,
Meta object at 0x12e892350>,
object at 0x12e892690>,
rtyMeta object at 0x12e8929d0>,
opertyMeta object at 0x122cfb-
ropertyMeta object at
MoPropertyMeta object at
emeta.MoPropertyMeta object at
MoPropertyMeta object at
meta.MoPropertyMeta object at

th all servers from
drop the code

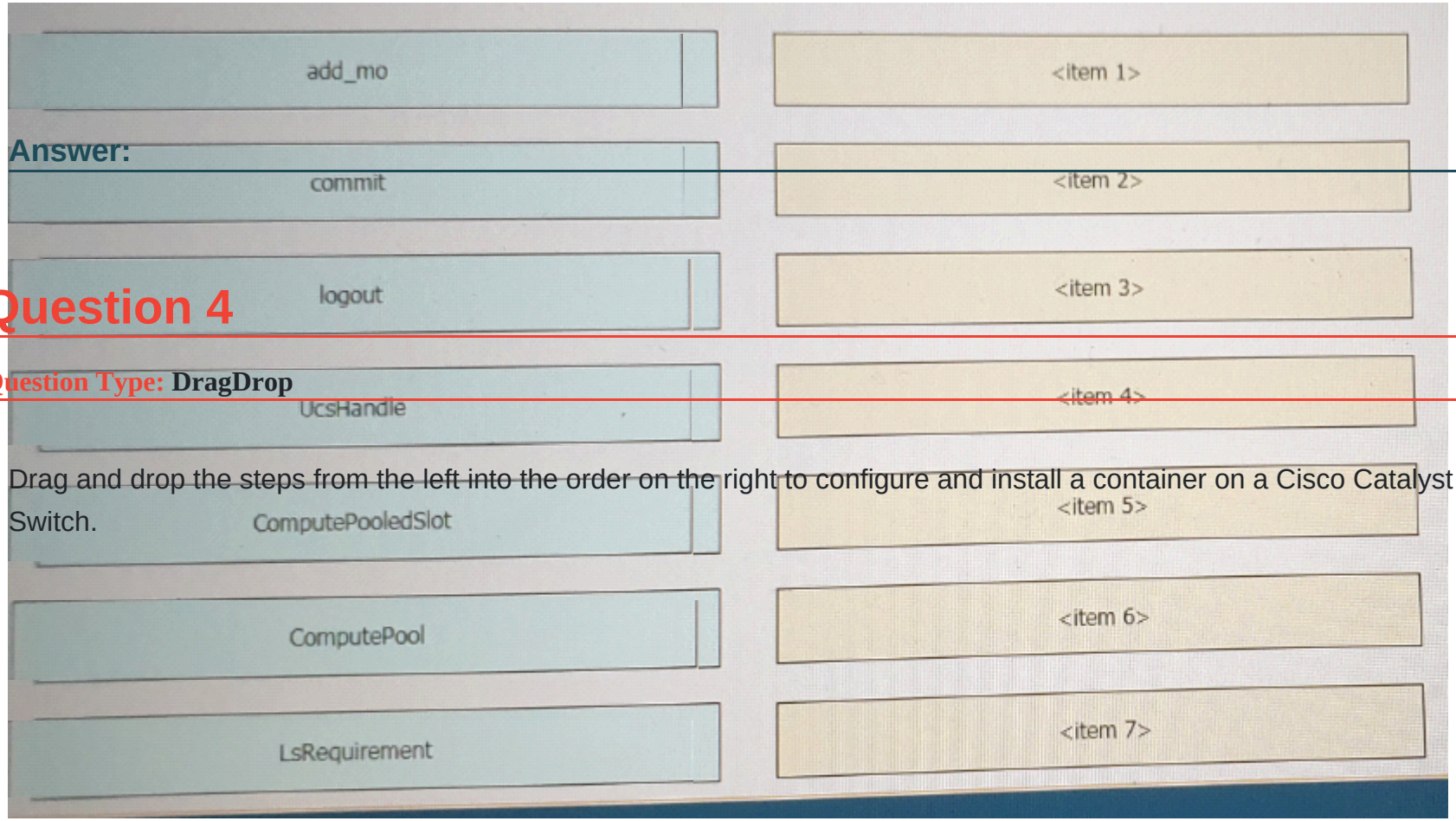
0x12e892a10>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892090>, 'restrict_migration': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892110>, 'pn_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e8926d0>, 'compute_ep_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfb350>}

Answer:

Question 4

Question Type: DragDrop

Drag and drop the steps from the left into the order on the right to configure and install a container on a Cisco Catalyst 9000 Series Switch.



cat9k# app-hosting start appid MYAPP	step 1
cat9k# app-hosting install appid MYAPP package flash:myapp.tar	step 2
cat9k# app-hosting activate appid MYAPP	step 3
cat9k(config)# iox	step 4

Answer:

Question 5

Question Type: FillInTheBlank

Fill in the blanks to complete the Python script to enable the SSID with a name of "371767916" in the network resource "11111111" using the Meraki Dashboard API.

Image not found or type unknown

Answer:

Explanation:

1. 371767916

2. 'PUT'

3. payload

Fill in the blanks to complete the Python script to enable the SSID with a name of "376699609" in the network resource "11111111"

```
import requests
url = "https://api.meraki.com/api/v0/11111111/ssids/"
payload = "{\r\n \"name\": \" 376699609 \",\r\n \"enabled\": true\r\n}"
headers = {
  'Accept': '*/*',
  'Content-Type': 'application/json'
}
response = requests.request( "POST" , url, headers=headers, data = payload )
print(response.text.encode('utf8'))
```

Question 6

Question Type: DragDrop

Resource Tab:

add_mo

commit

logout

UcsHandle

ComputePooledSlot

ComputePool

LsRequirement

ted

isconnecting

```
le import UcsHandle

n handle
handle = UcsHandle("192.168.1.1", "admin", "password")

handle.login()

# Logout from the server
handle.logout()
```

Refer UcsHandle API Reference for detailed parameter sets to `UcsHandle`

This module contains the general information for ComputePooledSlot ManagedObject.

add_mo

class

ucsmsdk.mometa.compute.ComputePooledSlot.**ComputePooledSlot**(*parent_mo_or_dn*, [source]

commit

Bases: **ucsmsdk.ucsmo.ManagedObject**

logout

consts = *<ucsmsdk.mometa.compute.ComputePooledSlot.ComputePooledSlot-*

UcsHandle

oMeta object>

naming_props = *set([u'chassisId', u'slotId'])*

ComputePooledSlot

', 'sacl': 'sacl', 'slotId': 'slot_id', 'assigned': 'assigned', 'owner': 'owner', 'prevAssignedToDn': 'prev_assigned_to_dn', 'child-

ComputePool

oolable_dn', 'chassisId': 'chassis_id', 'rn': }

prop_meta = *{'dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at*

LsRequirement

coremeta.MoPropertyMeta object at 0x1233ad5a0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad4d0>, 'assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392b10>, 'assigned': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392bd0>, 'owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad2d0>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad1d0>, 'poolable_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad350>, 'chassis_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392ad0>, 'slot_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad550>, 'prev_assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad3d0>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad450>}

utePool.**ComputePool**(*parent_mo_or_dn*, [source]

add_mo

Bases: **ucsmsdk.ucsmo.ManagedObject**

commit

consts = *<ucsmsdk.mometa.compute.ComputePool.ComputePoolConsts*

logout

mo_meta = *<ucsmsdk.ucscoremeta.MoMeta object>*

UcsHandle

us', 'policyLevel': 'policy_level', 'assignment-Order': 'assignment_order', 'sacl': 'sacl', 'policyOwner': 'policy_owner',

ComputePooledSlot

, 'childAction': 'child_action', 'name': 'name', 'descr': 'descr', 'rn': 'rn', 'size': 'size'}

ComputePool

remeta.MoPropertyMeta object at

scoremeta.MoPropertyMeta object at

0x1230ed3d0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at

LsRequirement

k.ucscoremeta.MoPropertyMeta object at 0x1230f8d90>, 'int_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed050>, 'assignment_order': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e10>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e90>, 'name': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed0d0>, 'descr': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8f10>, 'policy_owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed1d0>, 'policy_level': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed150>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed250>, 'size': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed350>}

add_mo

commit

logout

UcsHandle

ComputePooledSlot

ComputePool

LsRequirement

The SDK provides APIs to enable CRUD operations.

- Retrieve an object - query_dn,query_classid,query_dns,query_classids

The above APIs can be bunched together in a transaction (All or None). commit_mo commits the ch

All these methods are invoked on a UcsHandle instance. We refer it by handle in all the example

g to create a new handle.

1.5.3 Creating Objects

add_mo API.

Example:

The below example creates a new Service Profile(LsServer) Object under the parent org-root

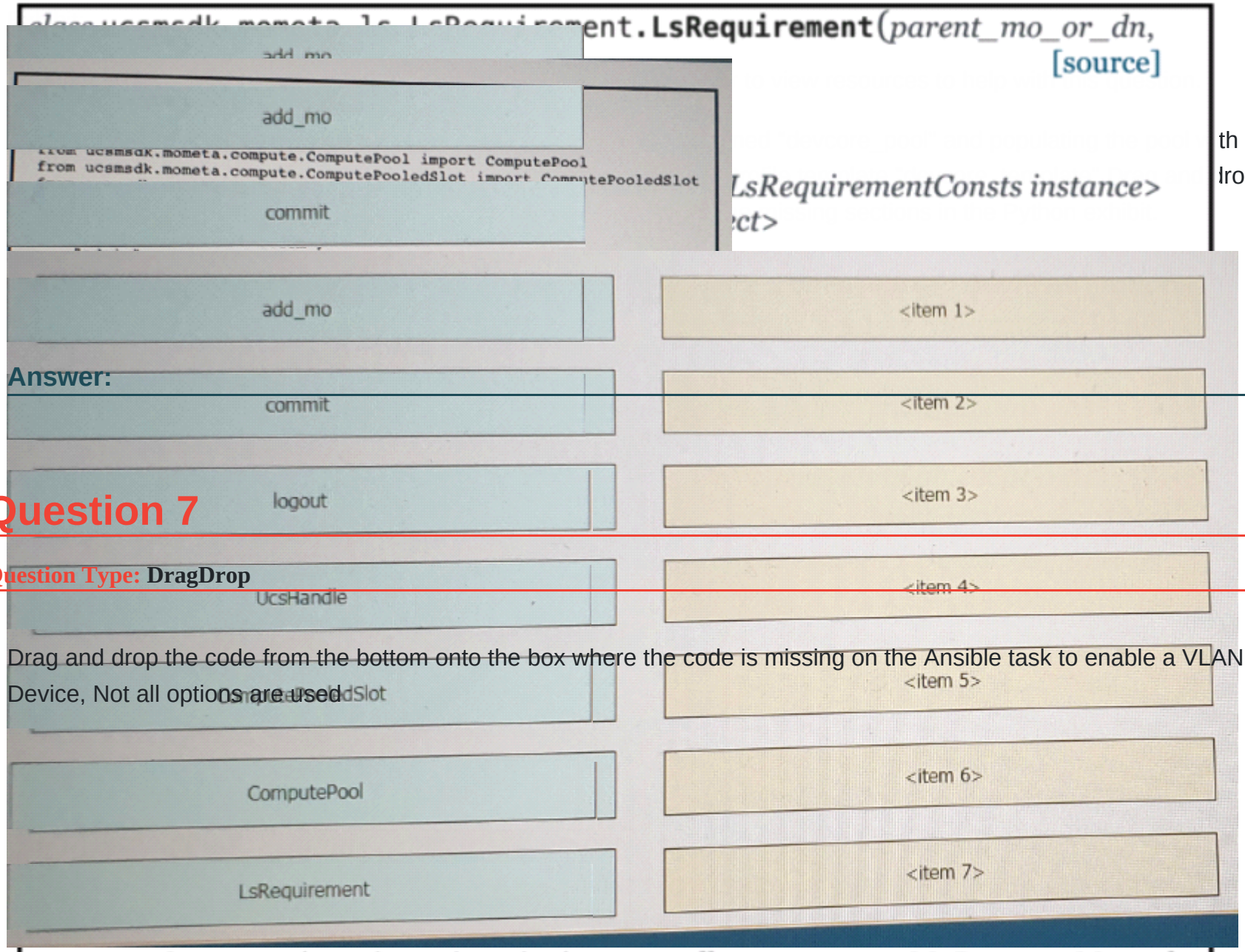
```
from ucsmsdk.mometa.ls.LsServer import LsServer

sp = LsServer(parent_mo_or_dn="org-root", name="sp_demo")
handle.add_mo(sp)
```

note: the changes will only be sent to server when handle.commit() is called.

Add Mo API reference

th all servers from
lrop the code



Answer:

Question 7

Question Type: DragDrop

Drag and drop the code from the bottom onto the box where the code is missing on the Ansible task to enable a VLAN on a Meraki MX Device, Not all options are used

0x12e892090>, 'restrict_migration': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892110>, 'pn_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e8926d0>, 'compute_ep_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfb350>}

Answer:

Question 8

Question Type: MultipleChoice

Refer to the exhibit.

Image not found or type unknown

The cURL POST request creates an OAuth access token for authentication with FDM API requests. What is the purpose of the file "@token_data" that cURL is handling?

Options:

A) This file is a container to log possible error responses in the request.

B) This file is given as input to store the access token received from FDM.

C) This file is used to send authentication related headers.

D) This file contains raw data that is needed for token authentication.

Answer:

D

Question 9

Question Type: **MultipleChoice**

Meraki Dashboard API Response

Response Status Code 200

Response Link Header

0000-0000-0000>; rel-first,

&startingAfter-Q2EK-3UBE-RRUY>; rel-next,

zzzz&perPage-3>; rel-last Response Body

{

'serial': 'Q2CV-V49B -RCMZ',

'mac': ''0c:8d:db:95:aa:39',

'networkid': 'L-566327653141846927',

'model11 : 'MV71',

'address': '430 E Cactus Ave .\nLas Vegas, NV 89183',

'lat': 36.00017,

'lng': -115.15302,

'notes': "',

'tagsn: '',,

'lanip: '192.168.0.25',

' configurationUpdatedAt': '2019-08-08T02:15:36Z', ' firmware11 : 'ca.rnera-3-3011

},

{

'ncune': 'Alex's MR84 - 1'1

'serial': 'Q2EK-2LYB-PCZP',

'mac': 'eO: 55:3d:10:56:8a', 'networkid': 'L 566327653141846927',

'model': 'MR84',

'address': ''11 ,

'lat': 39.9482993357826,

'lng': -82.9895675461739,

'notes': '',

'tags': ' ',

'lanip: null,

'configurationVpdatedAt': '2018-02-03Tll:02:37Z',

'firmware11 : 'Not running configured version''

},

{

'na.rne11 : 'Vegas Living Room MR84 11 ,

'serial': 'Q2EK-3UBE-RRUY',

'mac': 'e0:55:3d:10:5a:ca', 'networkid': 'L_566327653141846927' 1

'model': 'MR84',

'address': '430 E Cactus Ave.\nLas vegas, NV 89183', 'lat': 36.00015,

'lng': -115.15308,

'notes': ' ',

'tags': '11

1

'lanip: '192.168.0 .20',

'configurationVpdatedAt': '2018-09-29Tl2:23:21Z',

' firmware': 'Not running configured version'

Refer to the exhibit.

Image not found or type unknown

Which line of code must be added to this code snippet to allow an application to pull the next set of paginated items?

Options:

A) requests.get(url, links=['next']['url'])

B) requests.get(url, headers=links['next']['url'])

C) requests.get(res.links['next']['url'], headers=headers)

D) requests.get(res.headers.get('Link'')['next']['url'], headers=headers)

Answer:

C

Question 10

Question Type: **DragDrop**

requests.status_codes.codes.TOO_MANY_REQUESTS
requests.status_codes.codes.SERVER_ERROR
requests.status_codes.codes.TOO_MANY_REQUESTS
requests.status_codes.codes.SERVER_ERROR
requests.status_codes.codes.UNAUTHORIZED
try
Authorization
except Exception as e

<item 1>
<item 2>
<item 3>
<item 4>
<item 5>
<item 6>

```
():
  <item 1>:
      print(token)
sandboxdnac.cisco.com/dna/intent/api/v1/network-device"
                                                  aders=headers, data = payload)
        create_dnac_token()
    elif str(<item 4>) in str(e) and i < RETRIES:
        backoff_retry(get_dnac_devices)

def create_dnac_token():
    try:
        url = "https://sandboxdnac.cisco.com/dna/system/api/v1/auth/token"
        payload  = {}
        headers = {
        <item 3>: '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 str(<item 6>) in str(e):
            sys.exit("Server Unavailable")

def backoff_retry(func_retry):
    print("in backoff")
    time.sleep(backoff)
    backoff*=2
    func_retry()

if __name__ == "__main__":
    token = create_dnac_token()
    while True:
        print(get_dnac_devices())
```

Image not found or type unknown
Image not found or type unknown
Image not found or type unknown
Image not found or type unknown
Image not found or type unknown

the missing sections in the exhibit to

Answer:

Question 11

Question Type: DragDrop

Refer to the exhibit.

Drag and drop the steps from the left into the order of operation on the right for a successful OAuth2 three-legged authorization code grant flow.

Answer:

Question 12

Question Type: **FillInTheBlank**

Fill in the blanks to complete the Python script to enable the SSID with a name of "376699609" in the network resource "11111111" using the Meraki Dashboard API.

```
import requests
url = "https://api.meraki.com/api/v0/11111111/ssids/"
payload = "{\r\n \"name\": \" [          ] \",\r\n \"enabled\": true\r\n}"
headers = {
   'Accept': '*/*',
   'Content-Type': 'application/json'
}
response = requests.request( [          ] , url, headers=headers, data = [          ] )
print(response.text.encode('utf8'))
```

Answer: