

➤ **Vendor:** Cisco

➤ **Exam Code:** 350-901

➤ **Exam Name:** Developing Applications Using Cisco Core Platforms and APIs (DEVCOR)

➤ **New Updated Questions from** [Braindump2go](#) (Updated in [April/2021](#))

[Visit Braindump2go and Download Full Version 350-901 Exam Dumps](#)

QUESTION 159

A local Docker container with a Container ID of 391441516e7a is running a Python application. Which command is used to connect to a bash shell in the running container?

- A. `docker attach <Container ID>`
- B. `docker exec -it <Container ID> /bin/bash`
- C. `docker run -a stdin -a stdout <Container ID> /bin/bash`
- D. `docker container attach <Container ID>`

Answer: A

QUESTION 160

Refer to the exhibit. The JSON response is received from the Meraki location API. Which parameter is missing?

```
{
  "version": "3.0",
  "secret": "supersecret",
  "type": "WiFi",
  "data": {
    "networkId": "L 000000000000391274",
    "observations": [
      {
        "locations": [],
        "ipv4": null,
        "ssid": null,
        "os": null,
        "mac": "cc:cc:66:58:85:23",
        "latestRecord": [
          {
            "time": "2020-10-19T10:23:21z",
            "nearestApMac": "aa:aa:22:56:2e:42",
            "nearestApRssi": "-62"
          }
        ]
      }
    ]
  }
}
```

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

<https://www.braindump2go.com/350-901.html>

- A. apMac
- B. clientMac
- C. clientId
- D. accesspoint

Answer: B

Explanation:

<https://community.meraki.com/t5/Developers-APIs/Location-lat-lng-and-x-y-are-showing-similar-for-all-devices/td-p/65707>

QUESTION 161

Which two gRPC modes of model-driven telemetry are supported on Cisco IOS XE Software? (Choose two.)

- A. dial-in
- B. dial-out
- C. call-in
- D. call-out
- E. passive

Answer: AB

Explanation:

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1612/b_1612_programmability_cg/model_driven_telemetry.html#id_86392

QUESTION 162

Refer to the exhibit. Which parameter is missing from the JSON response to confirm the API version that is used?

```
1 {  
2   "version":  ,  
3   "secret": "supersecret",  
4   "type": "DevicesSeen",  
5   "data": {  
6     "apMac": "00:18:0a:13:dd:b0",  
7     "apFloors": [],  
8     "apTags": [  
9       "dev",  
10      "home",  
11      "test"  
12    ],  
13  }
```

- A. version 4
- B. v10
- C. 2
- D. version 2

Answer: C

QUESTION 163

What is a capability of the End User Monitoring feature of the AppDynamics platform?

- A. discovers traffic flows, nodes, and transport connections where network or application/network issues are developing
- B. monitoring local processes, services, and resource use, to explain problematic server

performance

- C. identifies the slowest mobile and IoT network requests, to locate the cause of problems
- D. provides metrics on the performance of the database to troubleshoot performance-related issues

Answer: C

Explanation:

<https://docs.appdynamics.com/display/PRO21/Overview+of+End+User+Monitoring>

QUESTION 164

Refer to the exhibit. Which command resolves the merge conflict by removing the previous commit from the commit history?

```
$ git checkout release-2.1
Switched to branch 'release-2.1'
Your branch is up to date with 'origin/release-2.1'.

$ git add -A
& git commit -m "Demo"
[release-2.1 6226cf6] Demo
1 file changed, 3 insertions(+)
$ git merge dev
Auto-merging python/mac.py
CONFLICT (content): Merge conflict in python/mac.py
Automatic merge failed; fix conflicts and then commit the result.
```

- A. `git checkout mac.py`
- B. `git reset --hard HEAD~1`
- C. `git rebase --abort`
- D. `git revert -m 1 HEAD`

Answer: B

QUESTION 165

What is the function of dependency management?

- A. separating code into modules that execute independently
- B. utilizing a single programming language/framework for each code project
- C. automating the identification and resolution of code dependencies
- D. managing and enforcing unique software version names or numbers

Answer: A

QUESTION 166

Refer to the exhibits. An interface named "GigabitEthernet2" has been configured on a Cisco IOS XE device. Using RESTCONF APIs as defined by the ietf-interfaces@2014-05-08.yang model, which two combinations of "rest_operation" and "payload" must be added to the Python script to set the "description" to "Configured by RESTCONF"? (Choose two.)

```
module: ietf-interfaces
+--rw interfaces
| +--rw interface* [name]
| +--rw name string
| +rw description? string
| +--rw type identityref
| +--rw enabled? boolean
| +--rw link-up-down-trap-enable? enumeration {if-mib}?
+--ro interfaces-state
+--ro interface* [name]
+--ro name string
+--ro type identityref
+--ro admin-status enumeration {if-mib}?
+--ro oper-status enumeration
+--ro last-change? yang:date-and-time
+--ro if-index int32 {if-mib}?
+--ro phys-address? yang:phys-address
+--ro higher-layer-if* interface-state-ref
+--ro lower-layer-if* interface-state-ref
+--ro speed? yang:gauge64
+--ro statistics
+--ro discontinuity-time yang:date-and-time
+--ro in-octets? yang:counter64
+--ro in-unicast-pkts? yang:counter64
+--ro in-broadcast-pkts? yang:counter64
+--ro in-multicast-pkts? yang:counter64
+--ro in-discards? yang:counter32
+--ro in-errors? yang:counter32
+--ro in-unknown-protos? yang:counter32
+--ro out-octets? yang:counter64
+--ro out-unicast-pkts? yang:counter64
+--ro out-broadcast-pkts? yang:counter64
+--ro out-multicast-pltas? yang:counter64
+--ro out-discards? yang:counter32
+--ro out-errors? yang:counter32
```

```
import requests
url = ("https://ios-xe-mgmt.cisco.com:9443/restconf/data/ietf-interfaces:" +
      "interfaces/interface=GigabitEthernet2")

headers = {
    'Accept': "application/yang-data+json",
    'Authorization': "Basic cm9vdDpEXlZheSFfMTAm",
    'Content-Type': "application"
}

response = requests.request(rest_operation, url, data=payload,
                             headers = headers, verify=False)

print (response.text)
```

- A.
- ```
rest_operation = "PATCH"

payload = " {\n \"ietf-interfaces:interface\": {\n \"name\": \"GigabitEthernet2\", \n \"description\": \"Configured by RESTCONF\" \n }\n}"
```
- B.
- ```
rest_operation = "PUT"

payload = " {\n    \"ietf-interfaces:interface\": {\n      \"name\": \"GigabitEthernet2\", \n      \"description\": \"Configured by RESTCONF\" \n    }\n}"
```
- C.
- ```
rest_operation = "PUT"

payload = "{\n \"ietf-interfaces:interface\": {\n \"name\": \"GigabitEthernet2\", \n \"description\": \"Configured by RESTCONF\", \n \"type\": \"iana-if-type:ethernetCsmacd\", \n \"enabled\" true, \n \"ietf-ip:ipv4\": {\n \"address\": [\n {\n \"ip\": \"10.255.255.1\", \n \"netmask\": \"255.255.255.0\" \n } \n] \n } \n } \n}"
```
- D.
- ```
rest_operation = "POST"

payload = " {\n    \"ietf-interfaces:interface\": {\n      \"name\": \"GigabitEthernet2\", \n      \"description\": \"Configured by RESTCONF\" \n    }\n}"
```
- E.
- ```
rest_operation = "POST"

payload = "{\n \"ietf-interfaces:interface\": {\n \"name\": \"GigabitEthernet2\", \n \"description\": \"Configured by RESTCONF\", \n \"type\": \"iana-if-type:ethernetCsmacd\", \n \"enabled\" true, \n \"ietf-ip:ipv4\": {\n \"address\": [\n {\n \"ip\": \"10.255.255.1\", \n \"netmask\": \"255.255.255.0\" \n } \n] \n } \n } \n}"
```

**Answer:** AB

#### QUESTION 167

Drag and Drop Question

Refer to the exhibit. Drag and drop the code snippets from the bottom onto the blanks in the code to provision a new

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

<https://www.braindump2go.com/350-901.html>



UCS server. Not all options are used.

```
class ucsm.sdk.mometa.ls.LsServer.LsServerConsts [source]
 ASSIGN_STATE_ASSIGNED= 'assigned'
 ASSIGN_STATE_FAILED= 'failed'
 ASSIGN_STATE_UNASSIGNED= 'unassigned'
 ASSOC_STATE_ASSOCIATED= 'associated'
 ASSOC_STATE_ASSOCIATING= 'associating'
 ASSOC_STATE_DISASSOCIATING= 'disassociating'
 ASSOC_STATE_FAILED= 'failed'
 ASSOC_STATE_UNASSOCIATED= 'unassociated'
 CONFIG_STATE_APPLIED= 'applied'
 CONFIG_STATE APPLYING= 'applying'
 CONFIG_STATE_FAILED_TO_APPLY= 'failed-to-apply'
 CONFIG_STATE_NOT_APPLIED= 'not-applied'
```

```
from ucsm.sdk.ucsevenhandler import UcsEventHandle
from ucsm.sdk.mometa.ls.LsServer import []

end_script = False

def _sp_associate_callback(mce):
 global end_script
 if mce.mo.assoc_state == LsServerConsts.ASSOC_STATE_ASSOCIATED:
 log.debug("SP:" + mce.mo.dn + " Assoc Successful. assoc_state: " +
 mce.mo.assoc_state)
 elif mce.mo.assoc_state == LsServerConsts.ASSIGN_STATE_FAILED:
 log.error("SP:" + mce.mo.dn + " Assoc Failed. assoc_state: " +
 mce.mo.assoc_state)
 end_script = True

def _sp_associate_monitor(event_handle, mo):
 [].add(managed_object=mo, prop= "assoc_state",
 success_value=[LsServerConsts.ASSOC_STATE_ASSOCIATED],
 failure_value=[LsServerConsts.ASSOC_ []],
 timeout_sec=600, call_back=_sp_associate_callback)
```

STATE\_ERROR]

STATE\_FAILED]

LsServerConsts

event\_handle

Answer:

```

from ucsm.sdk.ucsevenhandler import UcsEventHandle
from ucsm.sdk.mometa.ls.LsServer import LsServerConsts

end_script = False

def _sp_associate_callback(mce):
 global end_script
 if mce.mo.assoc_state == LsServerConsts.ASSOC_STATE_ASSOCIATED:
 log.debug("SP:" + mce.mo.dn + " Assoc Successful. assoc_state: " +
 mce.mo.assoc_state)
 elif mce.mo.assoc_state == LsServerConsts.ASSIGN_STATE_FAILED:
 log.error("SP:" + mce.mo.dn + " Assoc Failed. assoc_state: " +
 mce.mo.assoc_state)
 end_script = True

def _sp_associate_monitor(event_handle, mo):
 event_handle.add(managed_object=mo, prop="assoc_state",
 success_value=[LsServerConsts.ASSOC_STATE_ASSOCIATED],
 failure_value=[LsServerConsts.ASSOC_STATE_FAILED],
 timeout_sec=600, call_back=_sp_associate_callback)

```

STATE\_ERROR]

#### QUESTION 168

Drag and Drop Question

A Python application is being written to run inside a Cisco IOS XE device to assist with gathering telemetry data. Drag and drop the elements of the stack from the left onto the functions on the right to collect and display the telemetry streaming data.

|                        |                     |
|------------------------|---------------------|
| visualization platform | Cisco IOS XE device |
| data collector         | Elasticsearch       |
| data generator         | Kibana              |
| datastore              | Python application  |

Answer:

|  |                        |
|--|------------------------|
|  | data generator         |
|  | datastore              |
|  | visualization platform |
|  | data collector         |

#### QUESTION 169

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

<https://www.braindump2go.com/350-901.html>

**Drag and Drop Question**

Drag and drop the code from the bottom onto the box where the code is missing in the Python script to execute a REST API call to query all the NTP policy names and print the name of each policy. Not all options are used.

```
import requests, json
from intersight_auth import IntersightAuth

AUTH = IntersightAuth(
 secret_key_filename= '/tmp/secretfile.txt',
 api_key_id= 'api-key-id')
URL= 'https://www.intersight.com/api/v1/'

operations = [{"resource_path": " ",
 "request_method": "GET" }]

:

response = None
if operation['resource_path'] == "ntp/Policies":
 response = requests.get()

 = response.json()

for key, value in jsonResponse.items():
 if key = "Name":
 print(value)
```

for operation in operations

URL+operation  
['resource\_path'],auth=AUTH

jsonResponse

URL+operation[resource\_path],  
auth=api\_key\_id

ntp/Policies

response.json

for each operations

**Answer:**



```
import requests, json
from intersight_auth import IntersightAuth

AUTH = IntersightAuth(
 secret_key_filename= '/tmp/secretfile.txt',
 api_key_id= 'api-key-id')
URL= 'https://www.intersight.com/api/v1/'

operations = [{"resource_path": "ntp/Policies",
 "request_method": "GET" }]

for operation in operations :
 response = None
 if operation['resource_path'] == "ntp/Policies":
 response = requests.get(URL+operation
 ['resource_path'],auth=AUTH)

 jsonResponse = response.json()
 for key, value in jsonResponse.items():
 if key == "Name":
 print(value)
```

URL+operation[resource\_path],  
auth=api\_key\_id

response.json

for each operations

**QUESTION 170**

Drag and Drop Question

Drag and drop the descriptions from the left onto the related OAuth-defined roles on the right.

|                                                                            |                      |
|----------------------------------------------------------------------------|----------------------|
| provides access to a secured resource                                      | authorization server |
| user access tokens to accept and respond to secured resource requests      | client               |
| makes secured resource requests on behalf of the resource owner            | resource owner       |
| issues access tokens to the client after authenticating the resource owner | resource server      |

**Answer:**

|  |                                                                            |
|--|----------------------------------------------------------------------------|
|  | issues access tokens to the client after authenticating the resource owner |
|  | makes secured resource requests on behalf of the resource owner            |
|  | user access tokens to accept and respond to secured resource requests      |
|  | provides access to a secured resource                                      |

**QUESTION 171**

Drag and Drop Question

An engineer must access multiple bots that are running in an internal infrastructure. A different HTTPS URL is required for each bot. The infrastructure has just one public IP address and a Linux server with Apache installed. Drag and drop the actions from the left into the order of steps on the right to enable access to the bots inside. Not all options are used.

|                                                 |        |
|-------------------------------------------------|--------|
| Configure "Let's Encrypt" on the bot servers.   | step 1 |
| Enable a forward proxy in Apache.               | step 2 |
| Configure Apache virtual hosts.                 | step 3 |
| Enable a reverse proxy in Apache.               |        |
| Configure an Apache .htaccess file.             |        |
| Configure "Let's Encrypt" on the Apache server. |        |

**Answer:**

|                                                 |                                               |
|-------------------------------------------------|-----------------------------------------------|
| Enable a forward proxy in Apache.               | Configure Apache virtual hosts.               |
|                                                 | Configure "Let's Encrypt" on the bot servers. |
|                                                 | Enable a reverse proxy in Apache.             |
| Configure an Apache .htaccess file.             |                                               |
| Configure "Let's Encrypt" on the Apache server. |                                               |

**QUESTION 172**

Drag and Drop Question

A network engineer needs to retrieve interface data using the Cisco IOS XE YANG Model. Drag and drop the components from the bottom onto the box where the code is missing to complete the RESTCONF URI.

Not all options are used.

```
https:// {host}}:{{port}}/restconf/data/
```

:  /

|                     |               |
|---------------------|---------------|
| Cisco-native-IOS-XE | interface     |
| native              | Cisco-IOS-XE  |
| Cisco-IOS-XE-native | IOS-XE-native |

**Answer:**

`https:// {host}}:{{port}}/restconf/data/`  

`Cisco-IOS-XE-native` : `native` /  
`interface`

`Cisco-native-IOS-XE`

`Cisco-IOS-XE`

`IOS-XE-native`

**Explanation:**

[https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b\\_166\\_programmability\\_cg/restconf\\_prog\\_int.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b_166_programmability_cg/restconf_prog_int.html)