

DUMPS ARENA

Automating Cisco Collaboration Solutions (CLAUTO)

Cisco 300-835

Version Demo

Total Demo Questions: 10

Total Premium Questions: 60

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Network Programmability Foundation	11
Topic 2, Unified Communication	11
Topic 3, Cloud Collaboration	12
Topic 4, Collaboration Endpoints	12
Topic 5, Meetings	14
Total	60

QUESTION NO: 1

```
def my_function(listOfNumbers, b):
    total1 = b
    total2 = 0

    for num in listOfNumbers:
        if num < 5:
            total1 += num
        else:
            total2 += num

    return total1

listOfNums = [4, 3, 9, 1, 7, 2, 9, 0, 6, 8, 5]
a = my_function(listOfNums, 3)
print(a)
```

Refer to the exhibit. Which value of “a” is displayed when the Python code is executed?

- A. 10
- B. 13
- C. 18
- D. 21

ANSWER: B

QUESTION NO: 2

```
import requests

token = 'YzE1NTNiMjEtYzc0ZC00MjE2LWI5ZD_leb65fdf-9643-417f-9974-ad72cat0e10f'
room = 'Y2lzY29zcGFyazovL3VzLIJPT00BmNy0zY2NkLTg4Y2UtNDQwOWJmODBjNWEO'
```

Refer to the exhibit. The code includes the beginning of a short Python script that is constructed to notify the guard in case of an intruder alert. Which code snippet completes the script?

- ```
A. request.put('https://api.ciscospark.com/v1/rooms' + room,
 headers = {'Authorization': 'Bearer' + token },
 data = {
 'toPersonEmail' : 'guard2319@perimeter.net',
 'html' : 'Warning! Intruder alert in Section 12'
 }
)

B. request.post('https://api.ciscospark.com/v1/messages',
 headers = {'Authorization': 'Bearer' + token },
 data = {
 'roomId' : 'Perimeter Guard Space',
 'markdown' : '**Warning!** Intruder alert in Section 12'
 }
)

C. request.post('https://api.ciscospark.com/v1/ rooms' + room,
 headers = {'Authorization': 'Bearer' + token },
 data = {
 'text' : 'Warning! Intruder alert in Section 12'
 'markdown' : '**Warning!** Intruder alert in Section 12'
 }
)

D. request.post('https://api.ciscospark.com/v1/messages',
 headers = {'Authorization': 'Bearer' + token },
 data = {
 'toPersonEmail' : 'guard2319@perimeter.net',
 'markdown' : '**Warning!** Intruder alert in Section 12'
 }
)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**ANSWER: A**

### QUESTION NO: 3

How is a message created using Webex Teams REST APIs?

- A. POST https://api.ciscospark.com/v1/messages/{messageID}
- B. POST https://api.ciscospark.com/v1/messages/

- C. GET https://api.ciscospark.com/v1/messages/
- D. PUT https://api.ciscospark.com/v1/messages/

**ANSWER: B**

**QUESTION NO: 4**

What are two keys features of Cisco Unified Communications Manager Serviceability APIs? (Choose two.)

- A. save alarms and events for troubleshooting
- B. connect to serial console on phone hardware
- C. configure firewall rules on phone hardware
- D. scan servers and devices for malware
- E. real-time monitoring of components

**ANSWER: A E**

**QUESTION NO: 5 - (DRAG DROP)**

DRAG DROP

The Python script executes a REST API to retrieve information from the Cisco Meeting Server about the call with an identifier of "987654321". Drag and drop the code onto the snippet to complete the Python script. Not all options are used.

**Select and Place:**

Answer Area

```
import requests
url = 'https://myserver:445/api/v1/[]/987654321'
headers = {
 '[]': 'Basic YXBpOmRDdbG9lZDEyMyE='
}
response = requests.request('[]', url, headers = headers)
print(response.text)
```

call

GET

Authorization

Authentication

calls

PUT

POST

callId

ANSWER:

## Answer Area

```
import requests

url = 'https://myserver:445/api/v1/ /987654321'

headers = {
 '': 'Basic YXBpOmRDdbG9lZDEyMyE='
}

response = requests.request(' ', url, headers = headers)

print(response.text)
```

Explanation:

## QUESTION NO: 6

```
<?xml version="1.0"?>
<inboundDialPlanRules total="3">
 <inboundDialPlanRule id="eedeeca5-c73b-441e-bc5a-71d9b256180b">
 <domain>cms.heleus.lab</domain>
 <priority>100</priority>
 </inboundDialPlanRule>
 <inboundDialPlanRule id="647511a0-0c7d-484f-8796-ee52dd3128f1">
 <domain>space.heleus.lab</domain>
 <priority>2</priority>
 </inboundDialPlanRule>
 <inboundDialPlanRule id="2c5662ab-7e5a-45a8-a133-5f4d6c6cb509">
 <domain>cms.lab</domain>
 <priority>1</priority>
 </inboundDialPlanRule>
</inboundDialPlanRules>
```

Refer to the exhibit. Which API call made to Cisco Meeting Server removes the inbound dial plan rule matching cms.lab domain?

- A. DELETE to /api/v1/inboundDialPalnRules/ with body attribute id="2c5662ab-7e5a-45a8-a133-5f4d6c6cb509".
- B. DELETE to /api/v1/inboundDialPalnRules/2c5662ab-7e5a-45a8-a133-5f4d6c6cb509.
- C. PUT to /api/v1/inboundDialPalnRules/2c5662ab-7e5a-45a8-a133-5f4d6c6cb509 with body attribute delete="True".
- D. PUT to /api/v1/inboundDialPalnRules/2c5662ab-7e5a-45a8-a133-5f4d6c6cb509.

**ANSWER: A**

#### QUESTION NO: 7

Which two capabilities can be implemented in a custom application using the Cisco Unified IP Phone Services API? (Choose two.)

- A. Authenticate the phone to the network.
- B. Display corporate directory information.
- C. Play multicast messages.
- D. Upgrade phone firmware.
- E. Create new phone devices.

**ANSWER: C E**

#### QUESTION NO: 8

```
import xows
import asyncio

async def start(ip, usr, pw):
 async with xows.XoWSClient(ip, username=usr, password=pw) as client:
 async def callback(data, id_):
 print(f'Feedback {id}(Id {id_}): {data}')

 await client.wait_until_closed()

async def task():
 await start('10.10.10.1', 'admin', 'T357c45e')

asyncio.run(task())
```

Refer to the exhibit. This Python script uses the websocket-based, xAPI library “pyxows” to monitor and print event details to the console when users interact with UI Extension/In-Room Control

‘Action Buttons’ deployed to the Touch-10 interface of a Webex room device. Which two code snippets successfully capture such events? (Choose two.)

- A. `await client.xEvent(['Event', '**'], callback)`
- B. `await client.subscribe(['Event', 'ActionButton', 'Clicked'], callback)`
- C. `await client.subscribe(['Event', '**'], callback)`
- D. `await client.xEvent(['Event', 'UserInterface', 'Extensions', 'Panel', 'Clicked'], callback)`
- E. `await client.subscribe(['Event', 'UserInterface', 'Extensions', 'Panel', 'Clicked'], callback)`

**ANSWER: D E**

### QUESTION NO: 9

Which two functions are provided by the Java-based computer telephony applications API? (Choose two.)

- A. Provide call blocking and screening for applications.
- B. Provide analytics about Cisco Unified Communications Manager endpoints and users.
- C. Control and observe Cisco Unified Communications Manager phones.
- D. Route calls by using computer telephony integration ports and route points (virtual devices).
- E. Allow provisioning of Cisco Unified Communications Manager endpoints and users.

ANSWER: D E

QUESTION NO: 10 - (DRAG DROP)

DRAG DROP

This Python script automates the creation of a Webex Teams space and adds participants to the space. Drag and drop code on the snippet to complete the script. Not all options are used.

Select and Place:

Answer Area

```
import requests

head = { 'Content-Type': ' ',
 'Authorization': 'Bearer NWU4NjQ0YWUtNjYitMWMY_-417f-ae0e10f' }

res = requests.post(url = 'https://api.ciscospark.com/v1/rooms/',
 headers = head, json = { ' ': 'Populate members' })

spaceId = res.json()['id']

members = ['ToEnglish@webex.bot', 'ToSpanish@webex.bot']
for members in members:
 requests.post(url='https://api.ciscospark.com/v1/ ',
 headers = head,
 json = { 'roomId': spaceId, ' ': member})
```

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 

ANSWER:

## Answer Area

```

import requests

head = { 'Content-Type': ' application/json ',
 'Authorization': 'Bearer NWU4NjQ0YWUtNjYItMWMY_-417f-ae0e10f' }

res = requests.post(url = 'https://api.ciscospark.com/v1/rooms/',
 headers = head, json = { ' name ': 'Populate members' })

spaceId = res.json()['id']

members = ['ToEnglish@webex.bot', 'ToSpanish@webex.bot']
for members in members:
 requests.post(url='https://api.ciscospark.com/v1/ personEmail /',
 headers = head,
 json = { 'roomId': spaceId, ' /memberships ': member})

```

/memberships

person

title

/rooms

/spaces

personEmail

application/xml

application/json

name

/members

## Explanation: