300-835Q&As

Automating Cisco Collaboration Solutions (CLAUTO)

Pass Cisco 300-835 Exam with 100% Guarantee

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

https://www.pass2lead.com/300-835.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

Which Cisco Meeting Server REST API object resource is used to retrieve active call information?

A. /activecalls

B. /calls

C. /getactivecalls

D. /callProfiles

Correct Answer: B

QUESTION 2

What is a benefit of using Python virtual environments?

A. It isolates dependencies of every project from the system and each other.

B. It allows Python to differentiate between package versions.

C. It frees the developer from installing the project dependencies.

D. It puts dependent packages in a common site-packages directory.

Correct Answer: A

QUESTION 3

Which two statements describe advantages of consuming APIs with asynchronous versus synchronous requests? (Choose two.)

A. All Cisco APIs are designed to be invoked asynchronously.

B. APIs respond more quickly when invoked asynchronously.

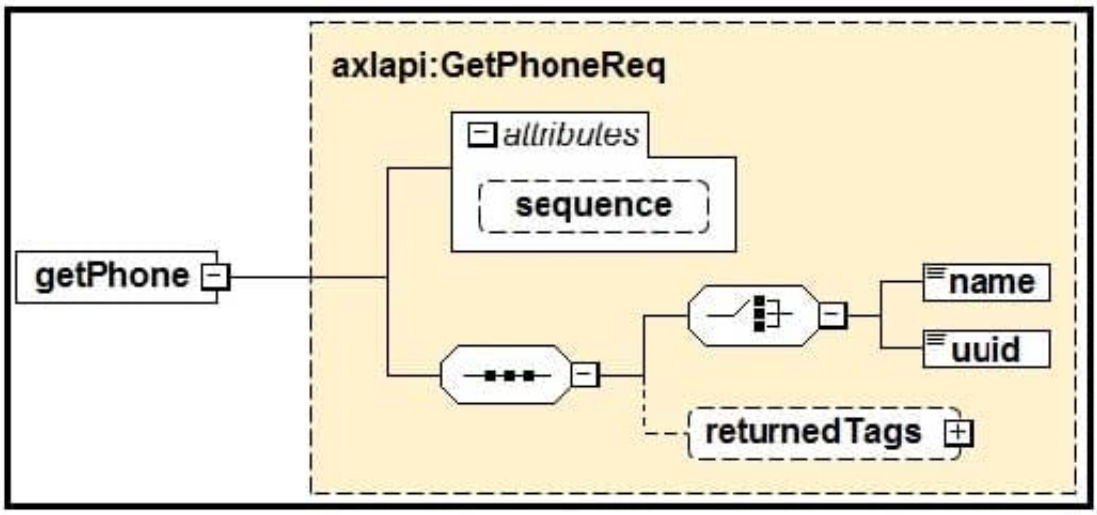
C. Asynchronous request coding is less complex.

D. Application threads do not block waiting for an asynchronous response.

E. Multiple asynchronous requests can be sent simultaneously.

Correct Answer: BD

QUESTION 4



Refer to the exhibit. Based on the schema diagram in the exhibit, which two XML objects are valid AXL request elements? (Choose two.)

A.
```
<soapenv:Body>
   <ns:getPhone sequence="1">
      <name>BOT209342098</name>
      <returnedtags><model/></returnedtags>
   </ns:getPhone>
</soapenv:Body>
```

B.
```
<soapenv:Body>
   <ns:getPhone>
      <name>BOT209342098</name>
      <returnedTags><model/></returnedTags>
   </ns:getPhone>
</soapenv:Body>
```

C.
```
<soapenv:Body>
   <ns:getPhone>
      <name>BOT209342098</name>
      <uuid>{2B1931A2-8FC2-A0C2-7282-B88B5A6356A0}</uuid>
   </ns:getPhone>
</soapenv:Body>
```

D.
```
<soapenv:Body>
   <ns:getPhone sequence="1">
      <uuid>{2B1931A2-8FC2-A0C2-7282-B88B5A6356A0}</uuid>
   </ns:getPhone>
</soapenv:Body>
```

E.
```
<soapenv:Body>
   <ns:getPhone sequence="1">
      <name>BOT209342098</name>
      <returnedTags>All</returnedTags>
   </ns:getPhone>
</soapenv:Body>
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Correct Answer: E

QUESTION 5

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})
```

/memberships	person	title	/rooms
/spaces	personEmail	application/xml	
application/json	name	/members	

Correct 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})
```

[person] [title] [/rooms]

[/spaces] [application/xml]

[/members]

QUESTION 6

DRAG DROP

An administrator is creating a script using the Python xAPI over WebSockets (pyxows) library. The goal of the script is to capture an event generated by a UI Extensions action button (former In-Room Control Panel). When the action button is clicked, the script displays an alert that says that the button ID was clicked. Drag and drop the code snippets into the locations to complete the script:

Select and Place:

Answer Area

```
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_):
            [                                                    ]
            [                                                    ]
        [                                                        ]
        await client.wait_until_closed()
async def task():
    [                                                            ]
[                                                                ]
```

```
await client.xCommand(['UserInterface','Message','Alert','Display'], Title=panelid, Text= "Was clicked")
panelId = data['Event']['UserInterface']['Extensions']['Panel']['Clicked']['PanelId']
await start('10.10.10.1','admin','T357c45e')
await client.subscribe(['Event','UserInterface','Extensions','Panel','Clicked'], callback, True)
asyncio.run(task())
```

Correct Answer:

Answer Area

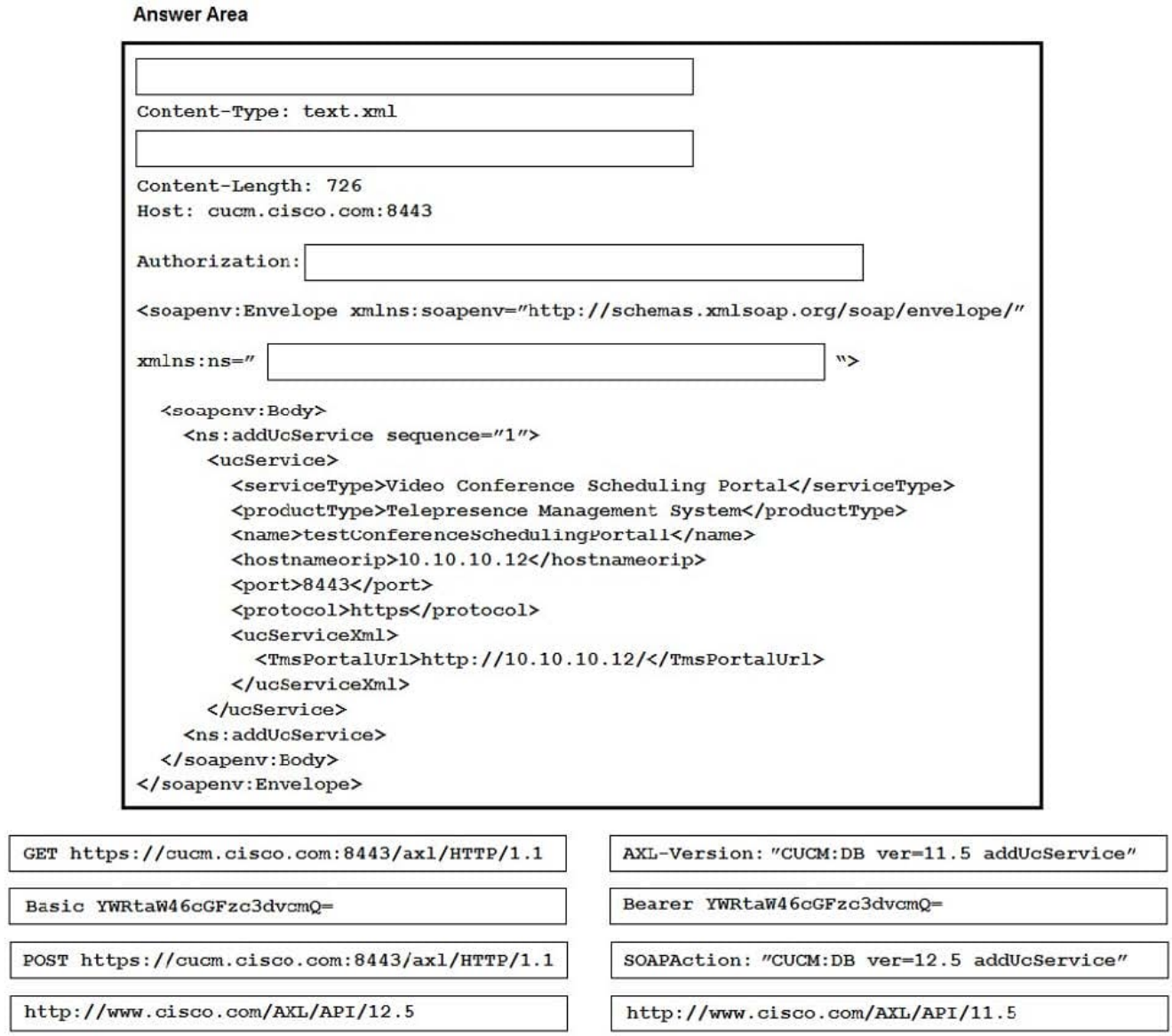
```
import xows
import asyncio

async def start(ip, usr, pw):
    async with xows.XoWSClient(ip,username=usr, password=pw) as client:
        async def callback(data,id_):
            panelId = data['Event']['UserInterface']['Extensions']['Panel']['Clicked']['PanelId']
            await client.xCommand(['UserInterface','Message','Alert','Display'], Title=panelid, Text= "Was clicked")
        await client.subscribe(['Event','UserInterface','Extensions','Panel','Clicked'], callback, True)
        await client.wait_until_closed()
async def task():
    await start('10.10.10.1','admin','T357c45e')
asyncio.run(task())
```

QUESTION 7

DRAG DROP

Drag and drop the code snippets to create a valid AXL API request. Not all options are used.

Select and Place:



Correct Answer:

Answer Area

```
POST https://cucm.cisco.com:8443/axl/HTTP/1.1
Content-Type: text.xml
SOAPAction: "CUCM:DB ver=12.5 addUcService"
Content-Length: 726
Host: cucm.cisco.com:8443

Authorization: Basic YWRtaW46cGFzc3dvcmQ=

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:ns=" http://www.cisco.com/AXL/API/12.5 ">

  <soapenv:Body>
    <ns:addUcService sequence="1">
      <ucService>
        <serviceType>Video Conference Scheduling Portal</serviceType>
        <productType>Telepresence Management System</productType>
        <name>testConferenceSchedulingPortal1</name>
        <hostnameorip>10.10.10.12</hostnameorip>
        <port>8443</port>
        <protocol>https</protocol>
        <ucServiceXml>
          <TmsPortalUrl>http://10.10.10.12/</TmsPortalUrl>
        </ucServiceXml>
      </ucService>
    <ns:addUcService>
  </soapenv:Body>
</soapenv:Envelope>
```

GET https://cucm.cisco.com:8443/axl/HTTP/1.1

AXL-Version: "CUCM:DB ver=11.5 addUcService"

Bearer YWRtaW46cGFzc3dvcmQ=

http://www.cisco.com/AXL/API/11.5

QUESTION 8

```
from pyxows import xows
import asyncio

async def main():

  async with xows.XoWSClient( '10.10.20.153', 'integrator', 'integrator' ) as client:

      async def callback( data, id ):
          print( f'Call status event: { data }' )

      [                                                                    ]

      await client.wait_until_closed()

asyncio.run( main() )
```

Refer to the exhibit. This exhibit is a simple Python scrip to monitor call activity for a local Webex room device. Which code snippet listens for call events and completes the script?

A. await client.xfeedback([`Status\\', `Call\\', `Status\\'], callback, False)

B. await client.subscribe([`Feedback\\', `Call\\', `Status\\'], callback, False)

C. await client.subscribe([`Status\\', `Call\\', `Status\\'], callback, False)

D. await client.xfeedback([`Feedback\\', `Call\\', `Status\\'], callback, False)

Correct Answer: C

QUESTION 9

A Webex Teams bot receives a Webhook payload, which notifies the bot that a message was created in a space. Which two API requests must be issued for the bot to answer the author of the message? (Choose two.)

A. POST /v1/messages

B. GET /v1/messages/{messageId}

C. GET /v1/webhooks/{webhookId}

D. PUT /v1/messages/{messageId}

E. POST /v1/webhooks

Correct Answer: CE

QUESTION 10

When the behavior of a Cisco collaboration device is customized, which use case requires an external control system because implementing JavaScript macro does not suffice?

A. Add a Join Webex meeting button to the touch panel.

B. Move the shutters up and down.

C. Trigger a "room-reset" to restore default configurations.

D. Implement an in-room control panel for speed-dialing.

Correct Answer: D