



300-835^{Q&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.geekcert.com/300-835.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**

```
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)`

Correct Answer: DE

QUESTION 2



```
<?xml version="1.0" encoding = "UTF-8"?>
<serv:message xmlns:serv="http://www.webex.com/schemas/2002/06/service"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <header>
    <securityContext>
      <siteName>apidemoeu</siteName>
      <webExID>alice@example.com</webExID>
      <sessionTicket>AAABb6TooBMAABUYA0gAKEgyU0sAAAAE9FpXDKRKO</sessionTicket>
    </securityContext>
  </header>
  <body>
    
  </body>
</serv:message>
```

Refer to the exhibit. Alice is a Webex Meetings admin for her organization. Which code snippet completes a Webex Meetings XML API request that returns all users?

- A. `<bodyContent xsi:type=<java:com.webex.service.binding.user.GetUser">`
`<webExID></webExID>`
`</bodyContent>`
- B. `<bodyContent xsi:type=<java:com.webex.service.binding.user.LstsummaryUser">`
`</bodyContent>`
- C. `<bodyContent xsi:type=<java:com.webex.service.binding.user.GetUser">`
`</bodyContent>`
- D. `<bodyContent xsi:type=<java:com.webex.service.binding.user.LstsummaryUser">`
`<webExID>*</webExID>`
`</bodyContent>`

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: A

QUESTION 3

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

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

await client.xCommand(['UserInterface', 'Message', 'Alert', 'Display'], Title=panelid, Text= "Was clicked")

await client.subscribe(['Event', 'UserInterface', 'Extensions', 'Panel', 'Clicked'], callback, True)

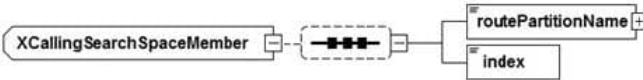
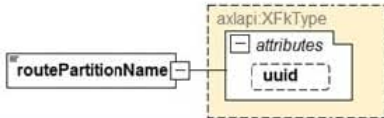

await client.wait_until_closed()

async def task():
    [redacted]
    [redacted]

asyncio.run(task())
```

QUESTION 4



complexType XCallingSearchSpaceMember						
diagram						
namespace	http://www.cisco.com/AXL/API/12.5					
children	routePartitionName index					
used by	elements XCss/members/member UpdateCssReq/removeMembers/member UpdateCssReq/addMembers/member UpdateCssReq/members/member					
source	<xsd:complexType name="XCallingSearchSpaceMember"> <xsd:sequence minOccurs="0"> <xsd:element name="routePartitionName" type="axlapi:XfkType" minOccurs="1" maxOccurs="1"/> <xsd:element name="Index" type="axlapi:XInteger" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>					
element XCallingSearchSpaceMember/routePartitionName						
diagram						
type	axlapi:XfkType					
properties	content complex					
attributes	Name	Type	Use	Default	Fixed	Annotation
	uuid	axlapi:XUUID	optional			
source	<xsd:element name="routePartitionName" type="axlapi:XfkType" minOccurs="1" maxOccurs="1"/>					
element XCallingSearchSpaceMember/routePartitionName						
diagram						
type	axlapi:XInteger					
properties	content simple					
attributes	Name	Type	Use	Default	Fixed	Annotation
	uuid	axlapi:XUUID	optional			
source	<xsd:element name="Index" type="axlapi:XInteger" minOccurs="1" maxOccurs="1"/>					

Refer to the exhibit. When a Calling Search Space is updated and a new member is added, how is a member defined?



- A. `<member uuid="{F6E0CD00-EC18-F16D-32E1-80C9A7A3D602}">
 <routePartitionName uuid="{5F2B2FFD-C7D2-4F30-9EF5-60110B56B939}">pt_translation-
 pattern_n</routePartitionName>
 <index>1</index>
</member>`
- B. `<member uuid="{F6E0CD00-EC18-F16D-32E1-80C9A7A3D602}">
 <routePartitionName>
 <uuid>"{5F2B2FFD-C7D2-4F30-9EF5-60110B56B939}"</uuid>
 <name>pt_translation-pattern_n</name>
 </routePartitionName>
 <index>1</index>
</member>`
- C. `<member>
 <routePartitionName uuid="{5F2B2FFD-C7D2-4F30-9EF5-60110B56B939}">pt_translation-
 pattern_n</routePartitionName>
 <index>1</index>
</member>`
- D. `<member">
 <routePartitionName uuid="{5F2B2FFD-C7D2-4F30-9EF5-60110B56B939}">
 <name>pt_translation-pattern_n</name>
 </routePartitionName>
 <index>1</index>
</member>`

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

QUESTION 5

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/>

Correct Answer: B

[300-835 PDF Dumps](#)

[300-835 Study Guide](#)

[300-835 Braindumps](#)