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.pass4itsure.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 type of endpoint is used by the Cisco Unified JTAPI implementation in Cisco Unified Communications Manager?

A. gateways

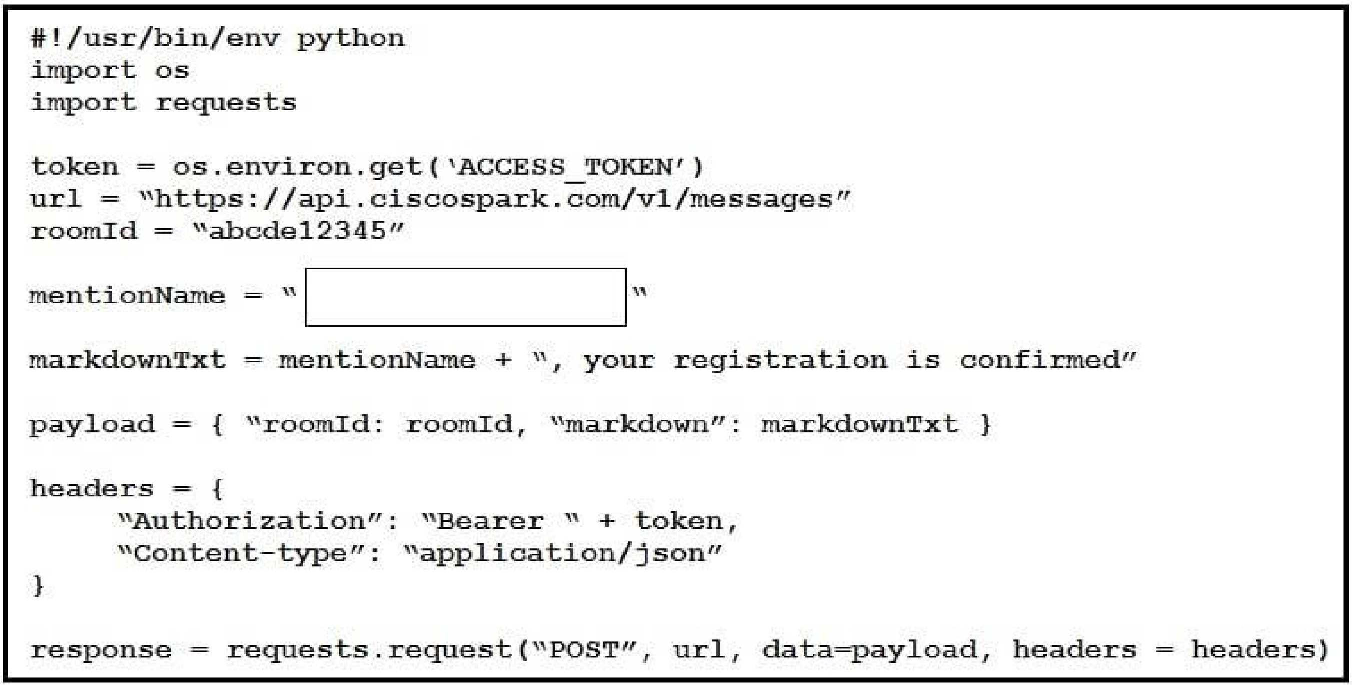
B. gatekeepers

C. phones

D. SIP trunks

Correct Answer: C

QUESTION 2



```
#!/usr/bin/env python
import os
import requests

token = os.environ.get('ACCESS_TOKEN')
url = "https://api.ciscospark.com/v1/messages"
roomId = "abcde12345"

mentionName = "                    "

markdownTxt = mentionName + ", your registration is confirmed"

payload = { "roomId: roomId, "markdown": markdownTxt }

headers = {
      "Authorization": "Bearer " + token,
      "Content-type": "application/json"
}

response = requests.request("POST", url, data=payload, headers = headers)
```

Refer to the exhibit. Which code snippet is used to mention the bot CiscoDevNet@webex.bot?

A.

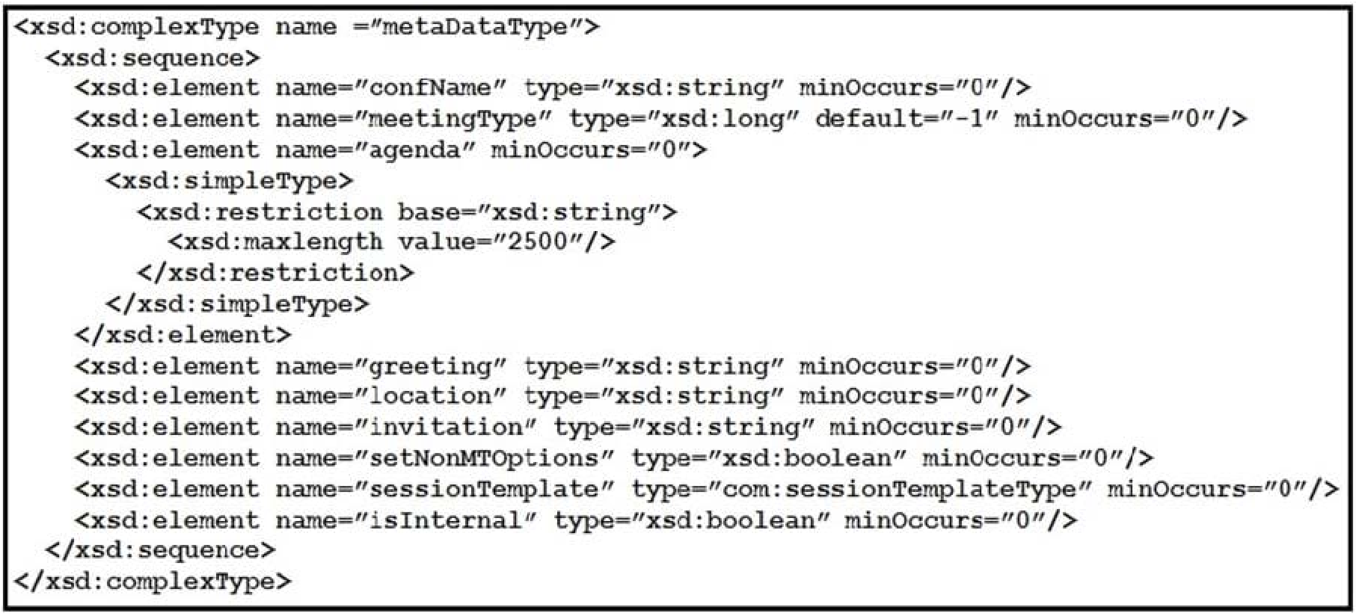
B. @CiscoDevNet@webex.bot

C. @personEmail.CiscoDevNet@webex.bot

D. (@personEmail.CiscoDevNet@webex.bot)

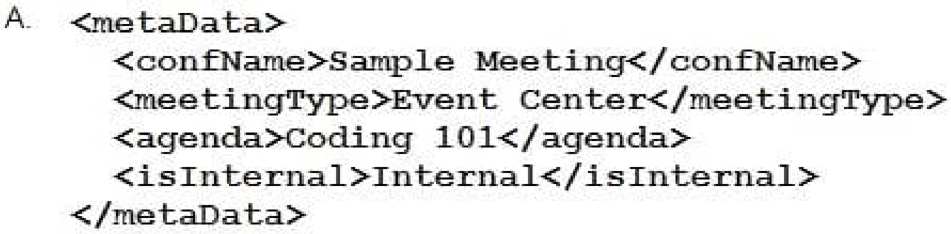
Correct Answer: B

QUESTION 3



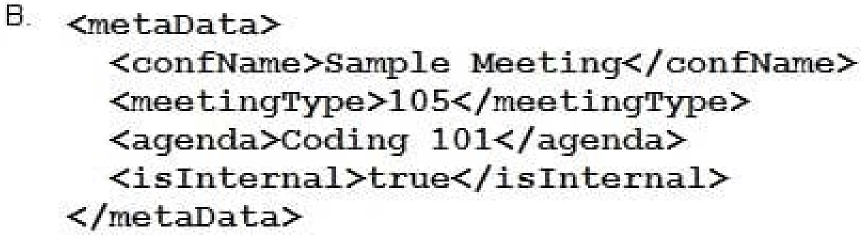
```
<xsd:complexType name =″metaDataType″>
  <xsd:sequence>
    <xsd:element name=″confName″ type=″xsd:string″ minOccurs=″0″/>
    <xsd:element name=″meetingType″ type=″xsd:long″ default=″-1″ minOccurs=″0″/>
    <xsd:element name=″agenda″ minOccurs=″0″>
      <xsd:simpleType>
        <xsd:restriction base=″xsd:string″>
          <xsd:maxlength value=″2500″/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name=″greeting″ type=″xsd:string″ minOccurs=″0″/>
    <xsd:element name=″location″ type=″xsd:string″ minOccurs=″0″/>
    <xsd:element name=″invitation″ type=″xsd:string″ minOccurs=″0″/>
    <xsd:element name=″setNonMTOptions″ type=″xsd:boolean″ minOccurs=″0″/>
    <xsd:element name=″sessionTemplate″ type=″com:sessionTemplateType″ minOccurs=″0″/>
    <xsd:element name=″isInternal″ type=″xsd:boolean″ minOccurs=″0″/>
  </xsd:sequence>
</xsd:complexType>
```

Refer to the exhibit. Based on this Webex Meetings XML API schema snippet, which XML object is a valid element?



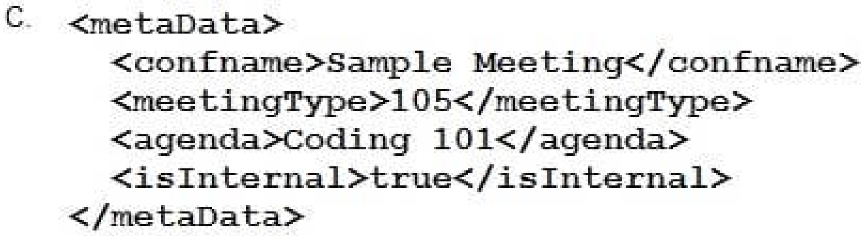
A.

```
<metaData>
   <confName>Sample Meeting</confName>
   <meetingType>Event Center</meetingType>
   <agenda>Coding 101</agenda>
   <isInternal>Internal</isInternal>
</metaData>
```



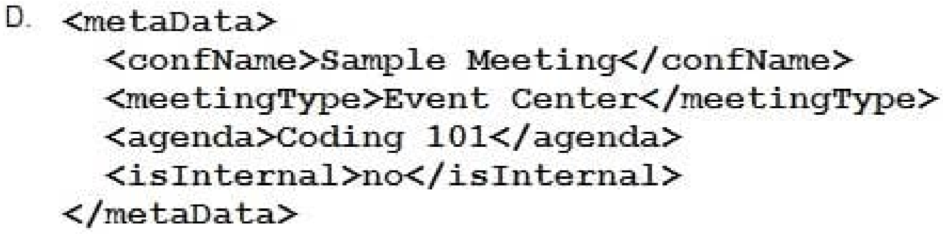
B.

```
<metaData>
   <confName>Sample Meeting</confName>
   <meetingType>105</meetingType>
   <agenda>Coding 101</agenda>
   <isInternal>true</isInternal>
</metaData>
```



C.

```
<metaData>
   <confname>Sample Meeting</confname>
   <meetingType>105</meetingType>
   <agenda>Coding 101</agenda>
   <isInternal>true</isInternal>
</metaData>
```



D.

```
<metaData>
   <confName>Sample Meeting</confName>
   <meetingType>Event Center</meetingType>
   <agenda>Coding 101</agenda>
   <isInternal>no</isInternal>
</metaData>
```

A. Option A

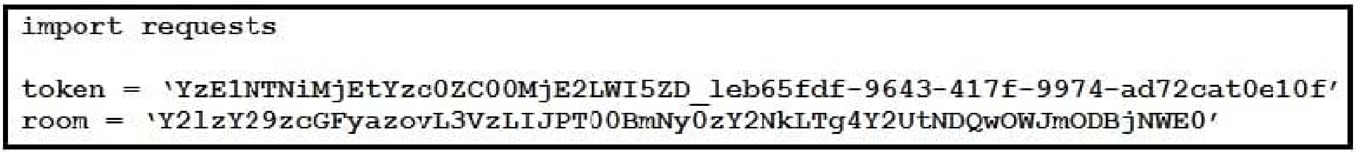
B. Option B

C. Option C

D. Option D

Correct Answer: C

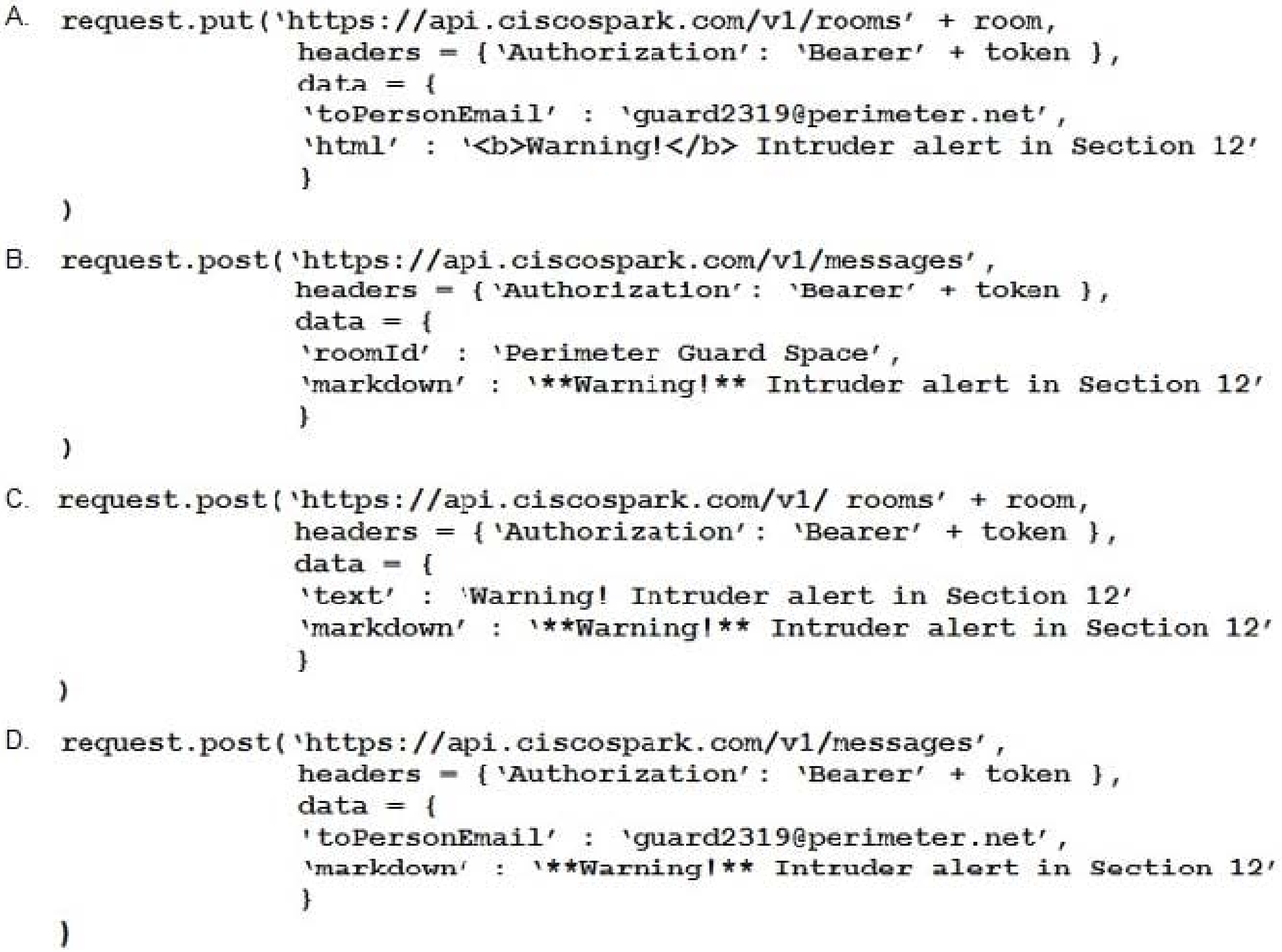
QUESTION 4



```
import requests

token = ‘YzE1NTNiMjEtYzc0ZC00MjE2LWI5ZD_leb65fdf-9643-417f-9974-ad72cat0e10f’
room = ‘Y2lzY29zcGFyazovL3VzLIJPT00BmNy0zY2NkLTg4Y2UtNDQwOWJmODBjNWE0’
```

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’ : ‘<b>Warning!</b> 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

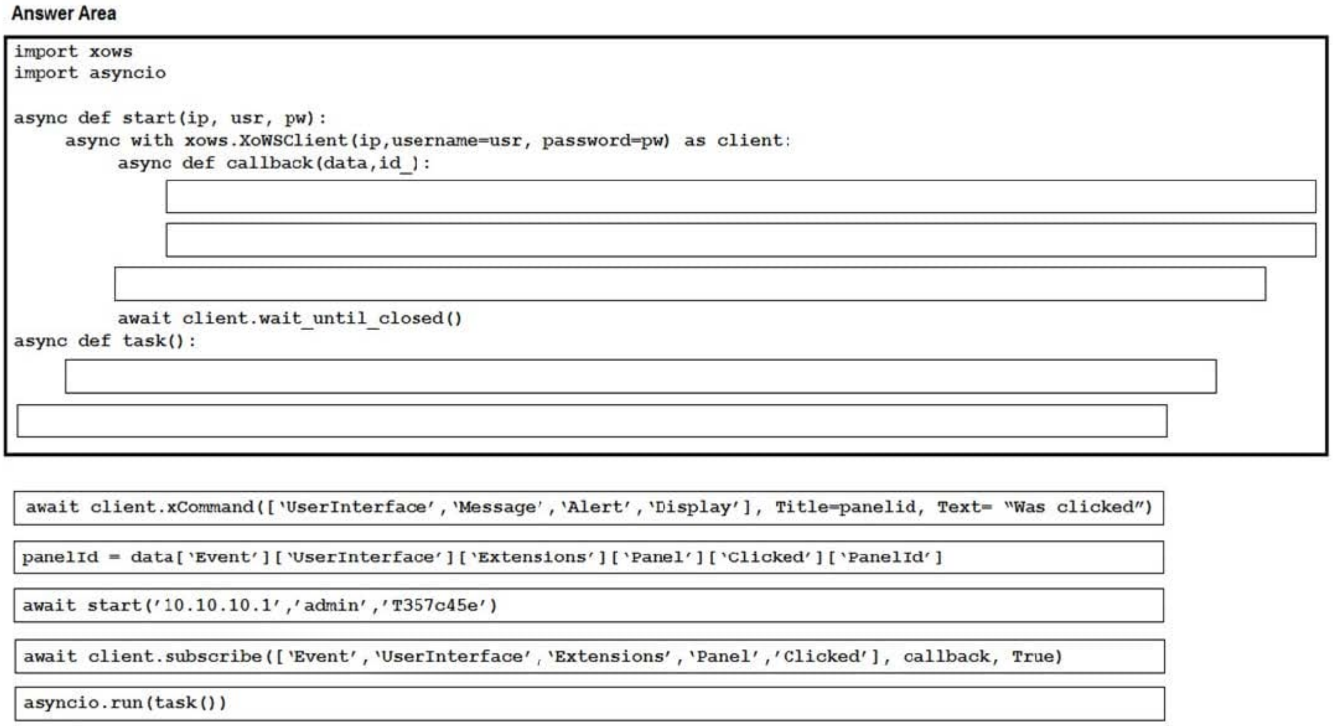
Correct Answer: A

QUESTION 5

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")
```

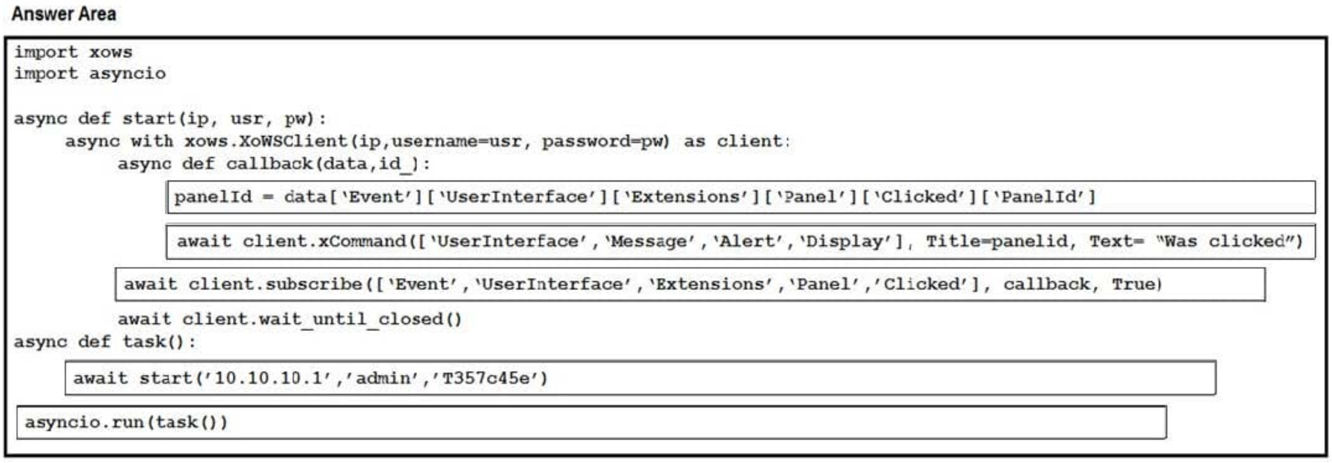
```
panelId = data['Event']['UserInterface']['Extensions']['Panel']['Clicked']['PanelId']
```

```
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())
```