

## Cisco.300-835.v2026-04-21.q46

<b>Exam Code:</b>	300-835
<b>Exam Name:</b>	Automating and Programming Cisco Collaboration Solutions
<b>Certification Provider:</b>	Cisco
<b>Free Question Number:</b>	46
<b>Version:</b>	v2026-04-21
<b># of views:</b>	111
<b># of Questions views:</b>	657
<a href="https://www.freecram.net/torrent/Cisco.300-835.v2026-04-21.q46.html">https://www.freecram.net/torrent/Cisco.300-835.v2026-04-21.q46.html</a>	

### NEW QUESTION: 1

Which interface is used to obtain performance counters from Cisco UCM?

- A. Performance Monitoring API
- B. Cisco Unified Reporting GUI
- C. Cisco Unified Serviceability GUI
- D. Cisco Unified Operating System Administration GUI

**Answer: (SHOW ANSWER)**

The Performance Monitoring API (Perfmon SOAP API) is the interface used to obtain performance counters from Cisco Unified Communications Manager (UCM). It allows programmatic access to metrics like CPU usage, memory, and call activity for performance monitoring.

### NEW QUESTION: 2

What is a benefit of using Python virtual environments?

- A. It puts dependent packages in a common site-packages directory.
- B. It allows Python to differentiate between package versions.
- C. It frees the developer from installing the project dependencies.
- D. It isolates dependencies of every project from the system and each other.

**Answer: (SHOW ANSWER)**

### NEW QUESTION: 3

What is a capability of the Cisco IP Phone Services API?

- A. It uses SOAP for all communication between clients and phones.
- B. It uses the XML schema file to define the behavior of the service.
- C. It bulk updates multiple phones with new firmware.
- D. It creates a DNS record in the server hosting for the IP Phone Services.

**Answer: (SHOW ANSWER)**

Explanation:

The Cisco IP Phone Services API enables developers to create custom services for Cisco IP phones. These services use XML to define how content is displayed and interacted with on the phone. The behavior of each service is structured based on Cisco's XML object schema, which defines how elements like menus, input fields, and text should behave and render on the phone display.

#### NEW QUESTION: 4

Which application component involved in conversational bots categorizes each request in the objective the user is trying to accomplish?

- A. entity recognition
- B. role classification
- C. intent classification
- D. domain classification

**Answer: (SHOW ANSWER)**

Intent classification is the component in a conversational bot that determines the user's objective - what action the user wants to perform (e.g., "schedule meeting", "check voicemail"). It categorizes the request so the bot can respond appropriately.

#### NEW QUESTION: 5

Refer to the exhibit. The beginning of the code is provided. Drag and drop the code snippets from the bottom onto the blanks in the Python script to create a new phone in Cisco UCM using the AXL API. Some options may be used more than once. Not all options are used.

Select and Place:

```
import requests
import os
def add_phone(**arg):
    payload = f'''<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.cisco.com/AXL/API/11.5">
```

## Answer Area

```
<soapenv:Header/>
<soapenv:Body>
  <ns:[ ] sequence="?">
    <phone ctiid="?">
      <name>{arg["phone_name"]}</name>
      <description>{arg["phone_desc"]}</description>
      <product>{arg["phone_product"]}</product>
      <class>{arg["phone_class"]}</class>
      <protocol>{arg["protocol"]}</protocol>
      <protocolSide>{arg["protocol_side"]}</protocolSide>
      <devicePoolName uuid="?">{arg["device_pool_name"]}</devicePoolName>
      <lines>
        <line ctiid="?">
          <index>{arg["line_index"]}</index>
          <label>{arg["label_line"]}</label>
          <display>{arg["line_display"]}</display>
          <dirn ctiid="?">
            <pattern>{arg["pattern_line"]}</pattern>
            <routePartitionName uuid="?">{arg["route_partition_name"]}
            </routePartitionName>
          </dirn>
        </line>
      </lines>
      <phoneTemplateName uuid="?">{arg["phone_template_name"]}
    </phoneTemplateName>
  </ns:[ ]>
</soapenv:Body>
</soapenv:Envelope>'''
headers = { 'Content-Type': '[ ]', 'Authorization': 'Basic {auth}'
.format(auth=os.getenv("TOKEN")) }
requests.request("[ ]", "https://10.243.160.1:8443/axl/", headers=headers,
data=payload)
```

requests

text/xml

addPhone

PUT

POST

json/xml

newPhone

Answer:

## Answer Area

```
<soapenv:Header/>
<soapenv:Body>
  <ns:addPhone | sequence=?">
    <phone ctiid=?">
      <name>{arg["phone_name"]}</name>
      <description>{arg["phone_desc"]}</description>
      <product>{arg["phone_product"]}</product>
      <class>{arg["phone_class"]}</class>
      <protocol>{arg["protocol"]}</protocol>
      <protocolSide>{arg["protocol_side"]}</protocolSide>
      <devicePoolName uuid=?">{arg["device_pool_name"]}</devicePoolName>
      <lines>
        <line ctiid=?">
          <index>{arg["line_index"]}</index>
          <label>{arg["label_line"]}</label>
          <display>{arg["line_display"]}</display>
          <dirn ctiid=?">
            <pattern>{arg["patern_line"]}</pattern>
            <routePartitionName uuid=?">{arg["route_partion_name"]}</routePartitionName>
          </dirn>
        </line>
      </lines>
      <phoneTemplateName uuid=?">{arg["phone_template_name"]}</phoneTemplateName>
    </phone>
  </ns:addPhone | >
</soapenv:Body>
</soapenv:Envelope>'''
headers = { 'Content-Type': 'text/xml |', 'Authorization': 'Basic {auth}'
.format(auth=os.getenv("TOKEN")) }
requests.request('POST |', "https://10.243.160.1:8443/axl/", headers=headers,
data=payload)
```



requests | text/xml | addPhone | dirn | POST | text/xml | addPhone

### NEW QUESTION: 6

Drag and drop the code snippets from the bottom onto the blanks in the code to create a Cisco Webex Meetings meeting using the API. Not all options are used.

Select and Place:

## Answer Area

```
import requests, json

token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3NpdCIsImV4cCI6MTY0MjY0MjY0fQ.eyJ1b2wiOiJ1b2wiLCJ0eXBlIjoiYm9keSIsImV4cCI6MTY0MjY0MjY0fQ.yJbN8'

headers = {
    'Authorization': f'Bearer {token}',
    'Content-Type': 'application/json'
}

payload = {
    "title": "DevNet Webex Discussion",
    "password": "AGrtgl234",
    "start": "2020-11-10T15:00:00Z",
    "end": "2020-11-10T17:00:00Z",
    "enabledAutoRecordMeeting": False,
    "allowAnyUserToBeCoHost": False
}

response = requests.request("PUT",
    f'{BASE_URL}/v1/{meetingmkr}/meetings', headers=headers, data=json.dumps(payload))
print(response.text.encode('utf8'))
```

CISCO

Answer:



**Answer:**

- POST
- Content-Type: text/xml
- PUT
- putxml
- <BrandingConfig>
- </BrandingConfig>
- <Branding>
- postxml
- Content: text
- </Branding>
- <Configuration>
- </Configuration>

**Answer Area**

```
curl -X POST \
http://192.168.1.34/putxml \
-H 'Authorization: Basic aW50ZWdyYXRvcjppbnRlZ3JhdG9yDQo=' \
-H 'Content-Type: text/xml' \
-d '<Configuration>
<UserInterface>
  <CustomMessage>My Company Branding</CustomMessage>
</UserInterface>
PUT'
```

**NEW QUESTION: 8**

Drag and drop the code snippets from the bottom onto the blanks in the code to implement notifications on a room device. Not all options are used.

Select and Place:

**Answer Area**

```
var alertOn = false
var snapshotTrigger = false

function remoteMonitorTriggered() {
  _____
  snapshotTrigger = true;
  if(!alertOn){
    _____
    alertOn = true;
  }
}
```

- console.log('Snapshot event triggered');
- xapi.event.on('VideoSnapshotTaken', remoteMonitorTriggered);
- xapi.command('UserInterface Message Alert Display', {Text:Monitoring});
- xapi.command.event(remoteMonitorTriggered);
- xapi.event('VideoSnapshotTaken', remoteMonitorTriggered);
- console.logging('Snapshot event triggered');

Answer:

### Answer Area

```
var alertOn = false
var snapshotTrigger = false

function remoteMonitorTriggered() {
    console.log('Snapshot event triggered');
    snapshotTrigger = true;
    if(!alertOn){
        xapi.command('UserInterface Message Alert Display', {Text:Monitoring});
        alertOn = true;
    }
}

xapi.event.on('VideoSnapshotTaken', remoteMonitorTriggered);
```

- `console.log('Snapshot event triggered');`
- `xapi.event.on('VideoSnapshotTaken', remoteMonitorTriggered);`
- `xapi.command('UserInterface Message Alert Display', {Text:Monitoring});`
- `xapi.command.event(remoteMonitorTriggered);`
- `xapi.event('VideoSnapshotTaken', remoteMonitorTriggered);`
- `console.log('Snapshot event triggered');`

### NEW QUESTION: 9

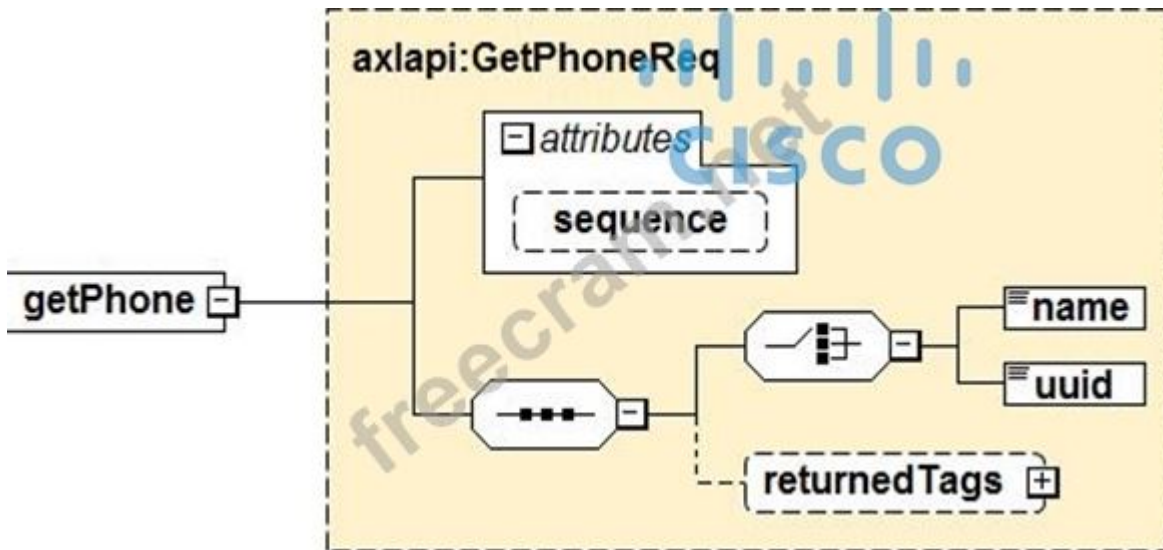
Which type of endpoint is used by the Cisco Unified JTAPI implementation in Cisco Unified Communications Manager?

- A. phones
- B. SIP trunks
- C. gateways
- D. gatekeepers

Answer: ([SHOW ANSWER](#))

### NEW QUESTION: 10

Refer to the exhibit.



Based on the schema diagram in the exhibit, which two XML objects are valid AXL <getPhone> request <Body> 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 B  
 B. Option D

- C. Option E
- D. Option A
- E. Option C

**Answer:** ([SHOW ANSWER](#))

**NEW QUESTION: 11**

What is the expiration time for a Cisco Webex Meeting API key to access the Webex API as an anonymous user?

- A. 6 months
- B. 12 hours
- C. one week
- D. 3 months

**Answer:** ([SHOW ANSWER](#))

**NEW QUESTION: 12**

Which type of token is required for authenticating requests to the Webex Meetings REST API?

- A. API token
- B. client token
- C. access token
- D. refresh token

**Answer:** ([SHOW ANSWER](#))

The Webex Meetings REST API requires an access token for authentication. This token is obtained through the Webex OAuth 2.0 flow and must be included in the Authorization header of API requests in the form:

Authorization: Bearer <access\_token>

**NEW QUESTION: 13**

Which two scenarios could be automated by combining the Webex Meetings XML API with other API-enabled systems? (Choose two.)

- A. reassigning an employee's scheduled Webex meeting to their manager then they leave the company
- B. scheduling new-hire orientation meetings as part of an HR system workflow
- C. automatically launching the weekly Webex scrum meeting on Mondays at 9 AM
- D. triggering Webex meeting recording when the host disconnect from the meeting.
- E. muting users in a Webex meeting when their Cisco Jabber presence status transitions to Away

**Answer:** ([SHOW ANSWER](#))

**NEW QUESTION: 14**

Drag and drop the botkit functions from the left onto the correct actions on the right.

**Answer Area**

hears()	Adds a question to the thread and evaluate the answer.
ask()	Sends a message with no pause for response.
say()	Replies to an incoming message.
reply()	Listens for keyword, phrases, or patterns in messages from users.

**Answer:**

**Answer Area**

hears()	ask() Evaluate the answer.
ask()	say() response.
say()	reply() .
reply()	hears() erns in

**NEW QUESTION: 15**

# room analytics

Available only on room devices

## GET Status PeopleCount

http://192.168.1.32/getxml?location=/Status/RoomAnalytics/PeopleCount

### HEADERS

Authorization Basic cmVtb3Rlc3Vwc69ydDo2U582a0JDYnJ0DQo=

location /Status/RoomAnalytics/PeopleCount

Example Request

Status PeopleCount

```
import requests

url = "http://192.168.1.32/getxml?location=/Status/RoomAnalytics/PeopleCount"

payload={}
headers = {
    'Authorization': 'Basic cmVtb3Rlc3Vwc69ydDo2U582a0JDYnJ0DQo='
}
```

View More

Refer to the exhibit. Drag and drop the code snippets from the bottom onto the blanks in the Python script to monitor how many people are in Room 70 by using xAPI and RoomAnalytics. Not all options are used.

Select and Place:

## Answer Area

```
import requests
import os
import xml.etree.ElementTree as ET

def ():
    url = "http://{endpoint}/getxml?location=/Status/RoomAnalytics/PeopleCount".
    format(endpoint=os.getenv("ROOM_ID"))
    payload={}
    headers = {
        'Authorization': 'Basic {token}'.(token=os.getenv("TOKEN"))
    }
    response = requests.request("", url, headers=headers,
    data=payload,
    verify=False)
    responseXml = ET.fromstring()

    count_people_res = responseXml.find('RoomAnalytics').find('PeopleCount')
    count_people_res = int(count_people_res.find('Current').text)

    if count_people_res > 0:
        print('There is {count_people}'.format(count_people=count_people_res))
    else:
        print('There is nobody')
if __name__ == "__main__":
    count_people()
```

Answer:

## Answer Area

```
import requests
import os
import xml.etree.ElementTree as ET

def ():
    url = "http://{endpoint}/getxml?location=/Status/RoomAnalytics/PeopleCount".
    format(endpoint=os.getenv("ROOM_ID"))
    payload={}
    headers = {
        'Authorization': 'Basic {token}'.(token=os.getenv("TOKEN"))
    }
    response = requests.request("", url, headers=headers,
    data=payload,
    verify=False)
    responseXml = ET.fromstring()

    count_people_res = responseXml.find('RoomAnalytics').find('PeopleCount')
    count_people_res = int(count_people_res.find('Current').text)

    if count_people_res > 0:
        print('There is {count_people}'.format(count_people=count_people_res))
    else:
        print('There is nobody')
if __name__ == "__main__":
    count_people()
```

**NEW QUESTION: 16**

Refer to the exhibit.

```
import requests

def createUser(apiKey, email, firstName, lastName, displayName):
    url = "https://api.ciscopark.com/v1/people"
    headers = {
        'Authorization': "Bearer " + apiKey,
        'Content-Type': "application/json"
    }
    body = {
        
    }
    response = requests.post(url = url, headers = headers, json=body)
    return response
```

Which snippet of code does a user with the administrator role use in the missing "body" section to create a new user in a Webex Teams organization?

- A. `"displayName": displayName,`  
`"firstName": firstName,`  
`"lastName": lastName`
- B. `"emails": {`  
`email`  
`},`  
`"displayName": displayName,`  
`"firstName": firstName,`  
`"lastName": lastName`
- C. `"email": email,`  
`"displayName": displayName,`  
`"firstName": firstName,`  
`"lastName": lastName`
- D. `"emails": [`  
`email`  
`],`  
`"displayName": displayName,`  
`"firstName": firstName,`  
`"lastName": lastName`

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

Answer: ([SHOW ANSWER](#))

**Valid 300-835 Dumps** shared by ExamDiscuss.com for Helping Passing 300-835 Exam! ExamDiscuss.com now offer the **newest 300-835 exam dumps**, the ExamDiscuss.com 300-835 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com 300-835 dumps with Test Engine here:

<https://www.examdiscuss.com/Cisco/exam/300-835/premium/> (132 Q&As Dumps, **35%OFF**

Special Discount Code: **freecram**)

**NEW QUESTION: 17**

Refer to the exhibit. Drag and drop the code snippets from the bottom onto the blanks in the code to construct a cURL command that will create a new space with the name Annual Meeting. Not all options are used.

Select and Place:

**Answer Area**

```
curl -X 
https://cms.domain.local/api/v1/ \
-H 'Authorization: Basic YWRtaW46YzFzY28xMjM=' \
-H 'content-type: multipart/form-data' \
-F ' = '
```

PUT	POST
Spaces	coSpaces
Annual Meeting	name

Answer:

## Answer Area

```
curl -X  \
  https://cms.domain.local/api/v1/ \
  -H 'Authorization: Basic YWRtaW46YzFzY28xMjM=' \
  -H 'content-type: multipart/form-data' \
  -F '='
```

PUT	POST
Spaces	coSpaces
Annual Meeting	name

CISCO

### NEW QUESTION: 18

Refer to the exhibit.

```
<?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>
```

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: ([SHOW ANSWER](#))

## NEW QUESTION: 19

Refer to the exhibit.

```
POST/HTTP/1.1
Content-Type:application/json; charset=UTF-8
X-Spark-Signature:06edc2e2f6d0a9d504c3f96367687cae0dedefcb
TrackingID: CLIENT_bfe5e832-45a6-4b96-bfd7-8c36122c3c1e_t:45dc3765

{
  "id": "Y21zY29zcGFyazovL3VzL1dFQkhPT0svYmI5MGE5MDAtNjM4YyYzRiOThhMjk5",
  "name": "exam question",
  "targetUrl": "https://2615c331.ngrok.io",
  "resource": "messages",
  "event": "created",
  "filter": "roomId=Y21zY29zcGFyazovMGNmNi0xMwVhLTk1ODktNzMyYjA4ZjlhYzE1",
  "orgId": "Y21zY29zcGFyazovL3VzL09kZi05NjQzLTQxN2YtOTk3NC1hZDcyY2F1MGUxMGY",
  "createdBy": "Y21zY29zcGFyazovL3VzL1lZTDiLTRkNjktOGYzOS0wNGI2YTFlNzE1ZmU",
  "appId": "Y21zY29zcGFy5YmE1MmZlY2JiMmM3ZDUxNWNiNGEwY2M5MwFh",
  "ownedBy": "creator",
  "status": "active",
  "created": "2019-11-22T07:08:51.934Z",
  "actorId": "Y21zY29zcGFyazovL3VzL3VLTTRhNDEtOGM0MS0yYWJkZjg5ZjQ0ZjQ",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzL01FU1mNy0xMwVhLTk1ODktNzMyYjA4ZjlhYzE1",
    "roomId": "Y21zY29zcGFyazovMGNmNi0xMwVhLTk1ODktNzMyYjA4ZjlhYzE1",
    "roomType": "group",
    "personId": "Y21zY29zcGFyazovhNDEtOGM0MS0yYWJkZjg5ZjQ0ZjQ",
    "personEmail": "user@example.com",
    "mentionedPeople": [
      "Y21zY29zcGFyazovL3VzLjktOGYzOS0wNGI2YIFlNzE1ZmU"
    ],
    "created": "2019-11-22T07:09:20.7002"
  }
}
```

A bot is receiving notifications such as the one displayed in the exhibit. Drag and drop the code onto the snippet to complete the API request that was sent to Webex Teams so that the bot will receive these notifications. Not all options are used.

Answer Area

```
curl -X  https://api.ciscopark.com/v1/  \
-H 
```

<input type="text" value='"secret": "VERY SECRET"'/>	<input "active"'="" "status":="" type="text" value='"actor": "/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="Authentication: Bearer"/&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;input type="text" value="Authentication: OAuth"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="GET"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="webhooks"/&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;input type="text" value="POST"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value='/>	<input type="text" value="messages"/>
<input type="text" value="events"/>	<input type="text" value="Authorization: Bearer"/>	<input type="text" value="PUT"/>
<input type="text" value="Authorization: OAuth"/>	<input "created",="" "event":="" "exam="" "filter":="" "https:="" "messages",="" "name":="" "resource":="" "roomid='Y2lzY29zcGFyazovMGNmNi0xMTVhLk1ODktNzMyYjA4ZjhhYzE1",' "secret":="" "targeturl":="" "very="" 'content-type:="" 'ntiwntuyowtndkz_pf84-9974-ad72ce10f'="" '{="" -d="" -h="" 2615c331.ngrok.io",="" <="" <input="" \="" application="" json'="" pre="" question",="" secret"'}="" type="text" value='"actor": "/&gt; }'/>	

<input type="text" value='"secret": "VERY SECRET"'/>	<input "active"'="" "status":="" type="text" value='"actor": "/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="Authentication: Bearer"/&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;input type="text" value="Authentication: OAuth"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="GET"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value="webhooks"/&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;input type="text" value="POST"/&gt;&lt;/td&gt;&lt;td&gt;&lt;input type="text" value='/>	<input type="text" value="messages"/>
<input type="text" value="events"/>	<input type="text" value="Authorization: Bearer"/>	<input type="text" value="PUT"/>
<input type="text" value="Authorization: OAuth"/>		

Drag and drop the code snippets from the bottom onto the boxes where the code is missing to listen for Call History events using the xAPI Python SDK.

Not all options are used.

Select and Place:

### Answer Area

```
import xows
import asyncio

async def start():
    async with xows. [ ] ('10.10.20.160', username='admin') as client:
        def callback(data, id_):
            print(f'Feedback (Id {id_}): {data}')

        await client. [ ] ([ ' [ ] ',
            ' [ ] ', 'Updated'], callback, True)

        await client.wait_until_closed()

asyncio.run(start())
```

- XAPIClient
- XoWSClient
- subscribe
- listen

- Event
- CallHistory
- Updated



Answer:

### Answer Area

```
import xows
import asyncio

async def start():
    async with xows. XoWSClient ('10.10.20.160', username='admin') as client:
        def callback(data, id_):
            print(f'Feedback (Id {id_}): {data}')

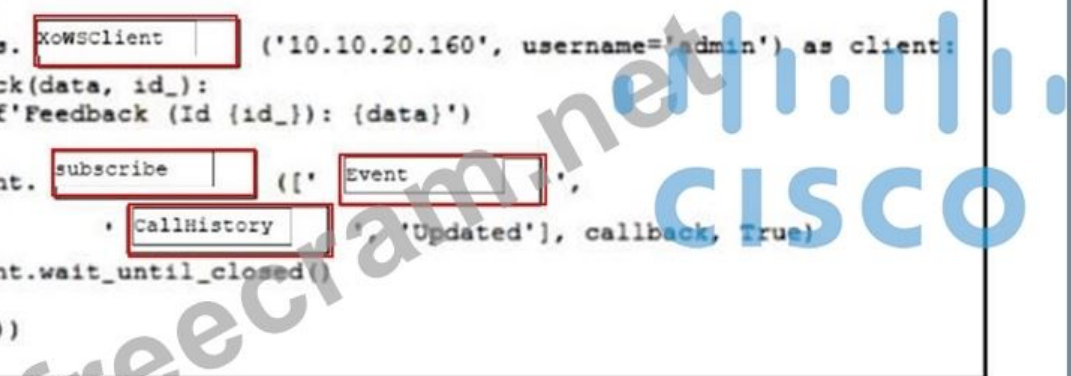
        await client. subscribe ([ ' Event ',
            ' CallHistory ', 'Updated'], callback, True)

        await client.wait_until_closed()

asyncio.run(start())
```

- XAPIClient
- XoWSClient
- subscribe
- listen

- Event
- CallHistory
- Updated



## NEW QUESTION: 21

Refer to the exhibit.



A developer has implemented ChatOps to a Webex Teams space as described in the exhibit. The Python script that pushes the notifications to the Teams space is shown. Drag and drop the code to complete the script. Not all options are used.

Answer Area

```
import requests, json, os

header = { [ ] : [ ] % os.environ.get("BOT_TOKEN"),
           "Content-Type": "application/json"}

message = "_The bot says: \n> I am a **developer** too!"

payload = {"roomId": os.environ.get("SPACE_ID"),
           "markdown": message}

res = requests.request([ ], url="https://api.ciscospark.com/v1/messages",
                       headers=header, data=[ ], verify=True)

if res.[ ] == 200:
    print("your message was successfully posted to Webex Teams")
else:
    print('failed with status code: %d' %res.status_code)

    if res.status_code == 404:
        print("please check the bot is in the space you're attempting to post to...")
    elif res.status_code == 400:
        print("please check the identifier of the space you're attempting to post to...")
    elif res.status_code == 401:
        print("please check if the access token is correct...")
```

json.dumps(payload)	"PUT"	status_code	payload
"Authentication"	"Authorization"	"Bearer"	status
"Bearer %s"	json(payload)	"POST"	code

Answer:

### Answer Area

```
import requests, json, os
header = { "Authorization" : "Bearer %s" % os.environ.get("BOT_TOKEN"),
          "Content-Type": "application/json"}
message = "_ The bot says:\n> I am a **developer** too!"
payload = {"roomId": os.environ.get("SPACE_ID"),
          "markdown": message}
res = requests.request("POST", url="https://api.discospark.com/v1/messages",
                      headers=header, data=payload, verify=True)
if res.status_code == 200:
    print("your message was successfully posted to Webex Teams")
else:
    print('failed with status code: %d' %res.status_code)

    if res.status_code == 404:
        print("please check the bot is in the space you're attempting to post to...")
    elif res.status_code == 400:
        print("please check the identifier of the space you're attempting to post to...")
    elif res.status_code == 401:
        print("please check if the access token is correct...")
```

json.dumps(payload)

"PUT"

status\_code

payload

"Authentication"

"Authorization"

"Bearer"

status

"Bearer %s"

json(payload)

"POST"

code

### NEW QUESTION: 22

Drag and drop the code snippets from the bottom onto the blanks in the script that sends the Server Down message to User One at userone@example.com and to the CLAUTONOC space with room ID A1bcA1 by using Cisco Webex. Not all options are used.

Select and Place:

## Answer Area

```
import requests
BASE_URL = "https://webexapis.com"
MESSAGES_URL = "/v1/messages"
TOKEN = '<TOKEN>'
USER = 'userone@example.com'
ROOM = 'AlbcA1_-'

headers = {'': 'Bearer ' + TOKEN,
           '': 'application/json'}
message1 = {'': USER, 'markdown': '**Server down**'}
message2 = {'': ROOM, 'markdown': "**Server Down**"}
response = requests.request('POST', BASE_URL + MESSAGES_URL,
                             headers=headers, json=message1)
response = requests.request('POST', BASE_URL + MESSAGES_URL,
                             headers=headers, json=message2)
```

Authorization

userEmail

roomId

X-Auth-Token

Content-Type

toPersonEmail

spaceId

Answer:

## Answer Area

```
import requests
BASE_URL = "https://webexapis.com"
MESSAGES_URL = "/v1/messages"
TOKEN = '<TOKEN>'
USER = 'userone@example.com'
ROOM = 'AlbcA1_-'

headers = {'Authorization': 'Bearer ' + TOKEN,
           'Content-Type': 'application/json'}
message1 = {'toPersonEmail': USER, 'markdown': '**Server down**'}
message2 = {'roomId': ROOM, 'markdown': "**Server Down**"}
response = requests.request('POST', BASE_URL + MESSAGES_URL,
                             headers=headers, json=message1)
response = requests.request('POST', BASE_URL + MESSAGES_URL,
                             headers=headers, json=message2)
```

Authorization

userEmail

roomId

X-Auth-Token

Content-Type

toPersonEmail

spaceId

### NEW QUESTION: 23

Drag and drop the code snippets from the bottom onto the blanks in the code to automate a query to list all phones on a test Cisco UCM cluster using an AXL SOAP request. Not all options are used.

Select and Place:

### Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.cisco.com/AXL/API/12.5">
  <soapenv:Header/>
  <soapenv:Body>
    <[ ]>
      <searchCriteria>
        <[ ]>
      </searchCriteria>
      <returnedTags>
      </returnedTags>
    <[ ]>
  </soapenv:Body>
</soapenv:Header>
</soapenv:Envelope>
```

- |               |              |
|---------------|--------------|
| ns:listPhone  | name>?       |
| ns:getPhone   | /ns:getPhone |
| /ns:listPhone | phone>?      |

Answer:

### Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.cisco.com/AXL/API/12.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:listPhone [ ]>
      <searchCriteria>
        <name>? [ ]>
      </searchCriteria>
      <returnedTags>
      </returnedTags>
    </ns:listPhone [ ]>
  </soapenv:Body>
</soapenv:Header>
</soapenv:Envelope>
```

- |               |              |
|---------------|--------------|
| ns:listPhone  | name>?       |
| ns:getPhone   | /ns:getPhone |
| /ns:listPhone | phone>?      |

**NEW QUESTION: 24**

Which Cisco Finesse API can control the communication between agents and servers?

- A. serviceability
- B. desktop
- C. configuration
- D. notifications

**Answer: (SHOW ANSWER)**

The Cisco Finesse desktop API is responsible for handling real-time communication between agents and servers, including call control, state changes, and agent actions on the desktop. It enables agents to interact with Cisco Finesse directly through browser-based or custom applications.

**NEW QUESTION: 25**

Drag and drop the code from the bottom onto the box where the JavaScript code is missing to list licenses in a critical state. Not all options are used.

Select and Place:

**Answer Area**

```
var unirest = require("unirest")
async function listLicenses(orgID){
var req = unirest("GET", "https://webexapis.com/v1/licenses")
req.query({
  "orgId": orgID,
});
req.headers({
  _____
});
req.end(function (response) {
  _____
  for ( x in licenses){
    if ( licenses[x]["consumedUnits"] > 0){
      console.log(licenses[x] _____)
    }
  }
});
}
```

- "Authorization": "Bearer 11111111AAAAaaa1111aaA",
- "Authentication": "11111111AAAAaaa1111aaAB22bb2B",
- var licenses = response.body["items"]
- var licenses = response.body["licenses"]
- ["name"]
- ["type"]



**Answer:**

## Answer Area

```
var unirest = require("unirest")
async function listLicenses(orgID){
  var req = unirest("GET", "https://webexapis.com/v1/licenses")
  req.query({
    "orgId": orgID,
  });
  req.headers({
    "Authorization": "Bearer 11111111AAAAaaaaa1111aaA",
  });
  req.end(function (response) {
    var licenses = response.body["items"]
    for ( x in licenses){
      if ( licenses[x]["consumedUnits"] > 0){
        console.log(licenses[x] ["name"]
      )
    }
  });
}
```

"Authorization": "Bearer 11111111AAAAaaaaa1111aaA",

"Authentication": "11111111AAAAaaaaa1111aaAB22bb2B",

var licenses = response.body["items"]

var licenses = response.body["licenses"]

["name"]

["type"]

### NEW QUESTION: 26

Using the Finesse REST APIs coupled with the Finesse notifications, which two use cases can be accomplished? (Choose two.)

- A. Develop a fully functioning custom agent desktop for chat, email, and SMS interactions including the content of the chat, email, and SMS.
- B. Add video calling capabilities to an existing application such as a CRM.
- C. Develop a secure monitoring application that captures the agent's states and dialog information of every agent in the contact center system.
- D. Add outbound call capabilities to an existing application such as a CRM.
- E. Develop a fully functioning custom agent desktop that has call control, workflows, and client logging.

Answer: ([SHOW ANSWER](#))

### NEW QUESTION: 27

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

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

Answer: ([SHOW ANSWER](#))

### NEW QUESTION: 28

Which two characteristics of REST APIs? (Choose two.)

- A. API operations for Create/Read/Update/Delete are mapped to standard HTTP methods.
- B. REST API extends Remote Procedure Call.
- C. The server manages the session state.
- D. Cookies are used for the duration of the session.
- E. Resources are accessed using Uniform Resource Identifiers.

Answer: ([SHOW ANSWER](#))

### NEW QUESTION: 29

Drag and drop the code snippets from the bottom onto the blanks in the Python script to create a new coSpace on the Cisco Meeting Server. Not all options are used.

Select and Place:

#### Answer Area

```
import   
from dotenv import load_dotenv  
  
import os  
global token  
token = os.environ.get("api-token")  
  
def create_cospace(cms_ip:str, name:str,uri:str, call_id:str ,callLegProfile:str,  
callProfile:str):  
    """Create a new cospace  
    """  
    global token  
    url = "https://{cms_ip}:445/api/v1/coSpaces".format(cms_ip=cms_ip)  
    payload = {  
        'name': name,  
        'uri':uri,  
        'callId':call_id,  
        'callLegProfile':callLegProfile,  
        'callProfile':callProfile,  
    }  
    headers = {  
         (token)'.format(token=token),  
        'Content-Type': 'application/x-www-form-urlencoded'  
    }  
    requests.request("", url,  
headers=headers, data=payload)
```

PUT

requests

'Basic Authorization:'

load\_dotenv()

POST

'Authorization': 'Basic'

getenv()

Answer:

## Answer Area

```
import requests
from dotenv import load_dotenv
load_dotenv()

import os
global token
token = os.environ.get("api-token")

def create_cospace(cms_ip:str, name:str,uri:str, call_id:str ,callLegProfile:str,
callProfile:str):
    """Create a new cospace
    """
    global token
    url = "https://{cms_ip}:445/api/v1/coSpaces".format(cms_ip=cms_ip)
    payload = {
        'name': name,
        'uri':uri,
        'callId':call_id,
        'callLegProfile':callLegProfile,
        'callProfile':callProfile,
    }
    headers = {
        'Authorization': 'Basic {token}'.format(token=token),
        'Content-Type': 'application/x-www-form-urlencoded'
    }
    requests.request("POST", url,
headers=headers, data=payload)
```

PUT

requests

'Basic Authorization:'

load\_dotenv()

POST

'Authorization': 'Basic

getenv()

### NEW QUESTION: 30

Refer to the exhibit.

```

<xsd:element name="associatedDevices" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Also known as controlled devices.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element name="device" type="axlapi:String50" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>Name of device.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="primaryExtension" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Only DNs associated with the associatedDevices of the User can be used as primary extension.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element name="pattern" type="axlapi:String255" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>DN or Pattern.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="routePartitionName" type="axlapi:String50" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Route Partition to which the DN is associated.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

The exhibit shows the XML schema to associate a device for the AddUser operation in AXL. Which XML snippet is correct to associate a device with name "SEP151515151515"?

- A. <associatedDevices>  
  <device name="SEP151515151515">  
</associatedDevices>
- B. <routePartitionName>  
  <device>SEP151515151515</device>  
</routePartitionName>
- C. <routePartitionName>  
  <device name="SEP151515151515">  
</routePartitionName>
- D. <associatedDevices>  
  <device>SEP151515151515</device>  
</associatedDevices>

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

Answer: ([SHOW ANSWER](#))

### NEW QUESTION: 31

Drag drop the code to create a valid AXL API script to create a new Route Partition, using the Python Zeep library. Not all options are used.

### Answer Area

```
from requests import Session
from requests import auth
from zeep import Client
from zeep.transports import Transport

session = Session()
session.auth = auth. [ ] ('Administrator', 'ciscopsdt')

client = Client(wsdl = ' [ ] ', transport = Transport(session=session))

service = client.create_service('{http://www.cisco.com/AXLAPIService/}AXLAPIBinding',
                                'https://ds-ucm1251.cisco.com [ ] ')

data = {
    'name': 'Secure Lab'
}

resp = service. [ ] (data)
```

:8443/axl/

AXLAPI.wsdl

AXLSoap.xsd

HTTPODigestAuth

HTTPBasicAuth

createPartition

addRoutePartition

/axlapi

### Answer:

#### Answer Area

```
from requests import Session
from requests import auth
from zeep import Client
from zeep.transports import Transport

session = Session()
session.auth = auth. HTTPBasicAuth ('Administrator', 'ciscopsdt')

client = Client(wsdl = ' AXLAPI.wsdl ', transport = Transport(session=session))

service = client.create_service('{http://www.cisco.com/AXLAPIService/}AXLAPIBinding',
                                'https://ds-ucm1251.cisco.com :8443/axl/ ')

data = {
    'name': 'Secure Lab'
}

resp = service. addRoutePartition (data)
```

:8443/axl/

AXLAPI.wsdl

AXLSoap.xsd

HTTPODigestAuth

HTTPBasicAuth

createPartition

addRoutePartition

/axlapi

**Valid 300-835 Dumps** shared by ExamDiscuss.com for Helping Passing 300-835 Exam!  
ExamDiscuss.com now offer the **newest 300-835 exam dumps**, the ExamDiscuss.com  
300-835 exam **questions have been updated** and **answers have been corrected** get the

newest ExamDiscuss.com 300-835 dumps with Test Engine here:

<https://www.examdiscuss.com/Cisco/exam/300-835/premium/> (132 Q&As Dumps, **35%OFF**

Special Discount Code: **freecram**)

### NEW QUESTION: 32

webexTeamsAPI		
	access_tokens	get() refresh()
	admin_audit_events	list()
	attachment_actions	create() get()
	events	list() get()
	guest_issuer	create()
	licenses	list() create()
	memberships	list() create() get() update() delete()
	messages	list() list_direct() create() get() delete()
	organizations	list() create()
	people	list() create() get() update() me()
	roles	list() create()
	rooms	list() create() get() get_meeting_info() update() delete()
	team_memberships	list() create() get() update() delete()
	teams	list() create() get() update() delete()
	webhooks	list() create() get() update() delete()

```
create(name, targetUrl, resource, event, filter=None, secret=None, **request_parameters) [source]
```

Create a webhook.

Parameters:

- `name (basestring)` - A user-friendly name for this webhook.
- `targetUrl (basestring)` - The URL that receives POST requests for each event.
- `resource (basestring)` - The resource type for the webhook.
- `event (basestring)` - The event type for the webhook.
- `filter (basestring)` - The filter that defines the webhook scope.
- `secret (basestring)` - The secret used to generate payload signature.
- `**request_parameters` - Additional request parameters (provides support for parameters that may be added in the future).

Returns:

A Webhook object with the details of the created webhook.

Return type:

Webhook

Raises:

- `TypeError` - If the parameter types are incorrect.
- `ApiError` - If the Webex Teams cloud returns an error.

Refer to the exhibit. Drag and drop the code snippets from the bottom onto the blanks in the code to construct a Python script that creates a webhook to notify the administrator when a message is deleted from a group Webex room. Not all options are used.

Select and Place:

```
from webexteamsdk import WebexTeamsAPI
Url = 'https://example.com/mywebhook'
Token = 'your_access_token'
api = WebexTeamsAPI(access_token=Token)
```

```
(
    name="My Webhook",
    targetUrl=Url,
    resource="messages",
    event="deleted",
    filter="roomId=group"
)
```

filter="roomId=group"

api.webhooks.create

filter="roomType=group"

resource="memberships"

resource="messages"

event="deleted"

Answer:

```
from webexteamsdk import WebexTeamsAPI
Url = 'https://example.com/mywebhook'
Token = 'your_access_token'
api = WebexTeamsAPI(access_token=Token)
```

```
api.webhooks.create
```

```
name="My Webhook",
targetUrl=Url,
```

```
resource="messages"
```

```
event="deleted"
```

```
filter="roomId=group"
```

filter="roomId=group"

api.webhooks.create

filter="roomType=group"

resource="memberships"

resource="messages"

event="deleted"

NEW QUESTION: 33

How is a message created using Webex Teams REST APIs?

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

Answer: ([SHOW ANSWER](#))

**NEW QUESTION: 34**

Drag and drop the elements to create the command to initiate a call to device@company.com using the Webex Devices xAPI SSH Interface. Not all options are used.

**Answer Area**

Command to initiate a call

: device@company.com

URI	Call
Number	xCommand
Device	xDial
Dial	Create

Answer:

**Answer Area**

Command to initiate a call

: device@company.com

URI	Call
Number	xCommand
Device	xDial
Dial	Create

**NEW QUESTION: 35**

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

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

**Answer: A,B (LEAVE A REPLY)**

**NEW QUESTION: 36**

Drag and drop the code snippets from the bottom onto the blanks in the code to configure a new cal bridge using the REST API. Not all options are used.

Select and Place:

**Answer Area**

```
import requests, json

token = 'eyJhbGc ... yJbNS'
url = 'http://cmsapi.cisco.io/api/v1/ [ ] '

headers = {
    'Authorization': f' [ ] (token)',
    'Content-Type': 'x-www-form-urlencoded'
}

body = "name=DevNet%20Bridges&address=https://cml.pod1.dnet&sipDomain=pod1.dnet"

response = requests.request(" [ ] ", url, headers=headers, data=body)
print(response.text. [ ] ('utf8'))
```

- callBridges
- encode
- POST
- Bearer
- manageBridge
- decode
- PUT

**Answer:**

## Answer Area

```
import requests, json
token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3NpdCIsImV4cCI6MTY1MjM0MjE1fQ.eyJ1cm91ciI6ImFkbWluIiwiaWF0IjoxNjUyMzQyMTUuMjE1fQ.'
url = 'http://cmsapi.cisco.io/api/v1/callBridges'
headers = {
    'Authorization': f'Bearer {token}',
    'Content-Type': 'x-www-form-urlencoded'
}
body = "name=DevNet%20Bridge&address=https://cml1.pod1.dnet&ipDomain=pod1.dnet"
response = requests.request("POST", url, headers=headers, data=body)
print(response.text.encode('utf8'))
```

callBridges

encode

POST

Bearer

manageBridge

decode

PUT

### NEW QUESTION: 37

What is a benefit of Python virtual environments?

- A. separate package dependencies for each application
- B. shared libraries across all projects
- C. shared workspace for Python applications
- D. dedicated CPU and memory for each application

**Answer:** ([SHOW ANSWER](#))

Explanation:

Python virtual environments allow each project to maintain its own isolated set of dependencies, avoiding conflicts between packages required by different applications. This is especially important when applications require different versions of the same library.

### NEW QUESTION: 38

Which format is used for guest tokens in Webex?

- A. OAuth2
- B. JWT
- C. OAuth1
- D. MD5

**Answer:** ([SHOW ANSWER](#))

Explanation:

Webex uses JWT(JSON Web Token) format for guest tokens to securely authenticate and authorize guest users in Webex applications without requiring full Webex accounts.

### NEW QUESTION: 39

Which Cisco UCM programming interface provides data that is needed to compile a report of call volume, trunk utilization, and cost and billing estimates?

- A. Perfmon SOAP API
- B. JTAPI
- C. CDRonDemandAPI
- D. RisPort70 SOAP API

**Answer: ([SHOW ANSWER](#))**

The CDRonDemand API provides access to Call Detail Records (CDRs) and Call Management Records (CMRs) from Cisco UCM. These records are essential for generating reports related to call volume, trunk utilization, and cost/billing estimates.

#### **NEW QUESTION: 40**

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

- A. /getactivecalls
- B. /callProfiles
- C. /calls
- D. /activecalls

**Answer: ([SHOW ANSWER](#))**

#### **NEW QUESTION: 41**

Drag and drop the code snippets into the correct order to create a Python Webex Teams REST API script that performs this sequence of tasks: create a Space, add a user to the Space, post the word "test" in the Space. Assume that the Python "requests" module has already been imported.

## Answer Area

Step 1

Step 2

Step 3

Step 4

Step 5

```
payload = { 'roomId': resp1.json()['id'], 'text': 'test' }  
resp2 = requests.post(url + '/messages', headers=headers, json=payload)
```

```
if resp2.status_code == 200:  
    print('Success!')
```

```
headers = { 'Content-type': 'application/json',  
            'Authorization': 'Bearer YTQzMzBiZmItNDJkNCO643-417974-ad72cae0e10f' }  
url = 'https://api.ciscospark.com/v1'
```

```
payload = { 'title': 'test' }  
resp1 = requests.post(url + '/rooms', headers=headers, json=payload)
```

```
payload = { 'roomId': resp1.json()['id'], 'personEmail': 'test@test.com' }  
requests.post(url + '/memberships', headers=headers, json=payload)
```

**Answer:**

## Answer Area

```
headers = { 'Content-type': 'application/json',  
'Authorization': 'Bearer YTQzMzBiZmItNDJkNCO643-417974-ad72cae0e10f' }  
url = 'https://api.ciscopark.com/v1'
```

```
payload = { 'roomId: resp1.json()['id'], 'personEmail': 'test@test.com' }  
requests.post(url + '/memberships', headers=headers, json=payload)
```

```
payload = { 'title': 'test' }  
resp1 = requests.post(url + '/rooms', headers=headers, json=payload)
```

```
payload = { 'roomId: resp1.json()['id'], 'text': 'test' }  
resp2 = requests.post(url + '/messages', headers=headers, json=payload)
```

```
if resp2.status_code == 200:  
print('Success!')
```

```
payload = { 'roomId: resp1.json()['id'], 'text': 'test' }  
resp2 = requests.post(url + '/messages', headers=headers, json=payload)
```

```
if resp2.status_code == 200:  
print('Success!')
```

```
headers = { 'Content-type': 'application/json',  
'Authorization': 'Bearer YTQzMzBiZmItNDJkNCO643-417974-ad72cae0e10f' }  
url = 'https://api.ciscopark.com/v1'
```

```
payload = { 'title': 'test' }  
resp1 = requests.post(url + '/rooms', headers=headers, json=payload)
```

```
payload = { 'roomId: resp1.json()['id'], 'personEmail': 'test@test.com' }  
requests.post(url + '/memberships', headers=headers, json=payload)
```

### NEW QUESTION: 42

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.

### Answer Area

```
import requests

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

headers = {
    '[ ]': 'Basic YXBpOmRDdbG91ZDEyMyE='
}

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

print(response.text)
```

- |  |   |
|--|---|
| <input type="text" value="call"/>          | <input type="text" value="GET"/>            |
| <input type="text" value="Authorization"/> | <input type="text" value="Authentication"/> |
| <input type="text" value="calls"/>         | <input type="text" value="PUT"/>            |
| <input type="text" value="POST"/>          | <input type="text" value="callId"/>         |

Answer:

### Answer Area

```
import requests

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

headers = {
    '[ ]': 'Basic YXBpOmRDdbG91ZDEyMyE='
}

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

print(response.text)
```

- |  |   |
|--|---|
| <input type="text" value="call"/>          | <input type="text" value="GET"/>            |
| <input type="text" value="Authorization"/> | <input type="text" value="Authentication"/> |
| <input type="text" value="calls"/>         | <input type="text" value="PUT"/>            |
| <input type="text" value="POST"/>          | <input type="text" value="callId"/>         |

**NEW QUESTION: 43**

Refer to the exhibit.

```
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)
```

Which value of "a" is displayed when the Python code is executed?

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

Answer: A ([LEAVE A REPLY](#))

**NEW QUESTION: 44**

Refer to the exhibit.

```
import requests

url = "https://webexapis.com/v1/devices?permissions=xapi"

payload = {}
headers = {
    "Content-Type": "application/json"
}

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

print(response.text)
```

An engineer needs to retrieve a list of devices that are xAPI enabled using Webex devices xAPIs with access token 111111111AAAaaaaaa1111aaA. To complete the python request, which line of code must be added to the box where the code is missing?

- A. 'Authorization: Token 111111111AAAaaaaaa1111aaA'
- B. 'Authorization: OAuth Token 111111111AAAaaaaaa1111aaA'
- C. 'Authorization: Bearer 111111111AAAaaaaaa1111aaA'
- D. 'Authorization: AccessToken 111111111AAAaaaaaa1111aaA'

Answer: ([SHOW ANSWER](#))

Explanation:

Webex APIs (including xAPI access through Webex Cloud) require the use of an OAuth 2.0 Bearer token in the Authorization header. The correct format is:

```
headers = {  
'Authorization': 'Bearer 111111111AAAaaaaaa1111aaA'  
}
```

This ensures the API call is authenticated correctly using the access token provided.

### NEW QUESTION: 45

Drag and drop the correct items from the right to complete this Python script to automate the creation of Cisco Webex Teams spaces and memberships. Not all options are used.

**Answer Area**

```
import requests  
  
def makePost(apiKey, url, body):  
    headers = {  
        '': "Bearer " + apiKey,  
        'Content-Type': "application/json"  
    }  
    response = requests.post(url = url, headers = headers, = body)  
    return response  
  
def createSpace(apiKey, roomName):  
    url = "https://api.ciscospark.com/v1/rooms/"  
    body = {  
        "title": roomName  
    }  
    response = makePost(apiKey, url, body)  
    roomId =   
    return roomId  
  
def addMembership(apiKey, roomId, membershipEmail):  
    url = "https://api.ciscospark.com/v1/memberships/"  
    body = {  
        "roomId": roomId,  
        "": membershipEmail  
    }  
  
    makePost(apikey, url, body)  
  
def main():  
    apiKey = input("What is your personal access token? ")  
    spacename = input("What is the name of the space you want to create? ")  
    membershipEmail = input("What is the email address of the person to add? ")  
    roomId = createSpace(apiKey, spacename)  
    addMembership(apiKey, roomId, membershipEmail)  
  
if __name__ == '__main__':  
    main()
```

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
-

Answer:

```
import requests

def makePost(apiKey, url, body):
    headers = {
        'Authorization': "Bearer " + apiKey,
        'Content-Type': "application/json"
    }
    response = requests.post(url = url, headers = headers, data = body)
    return response

def createSpace(apiKey, roomName):
    url = "https://api.ciscospark.com/v1/rooms/"
    body = {
        "title": roomName
    }
    response = makePost(apiKey, url, body)
    roomId = response.id
    return roomId

def addMembership(apiKey, roomId, membershipEmail):
    url = "https://api.ciscospark.com/v1/memberships/"
    body = {
        "roomId": roomId,
        "personEmail": membershipEmail
    }

    makePost(apikey, url, body)

def main():
    apiKey = input("What is your personal access token? ")
    spacename = input("What is the name of the space you want to create? ")
    membershipEmail = input("What is the email address of the person to add? ")
    roomId = createSpace(apiKey, spacename)
    addMembership(apiKey, roomId, membershipEmail)

if __name__ == '__main__':
    main()
```

response['id']

personEmail

json

data

Authorization

membershipEmail

apiKey

Authentication

OAuth

response.id

response[body]['id']

### NEW QUESTION: 46

Which component is used to integrate a webpage into the Cisco Finesse agent desktop?

- A. Finesse XMPP
- B. Finesse gadget
- C. Finesse REST API
- D. Finesse Python SDK

Answer: ([SHOW ANSWER](#))

A Finesse gadget is a web component (typically written in HTML/JavaScript) that is embedded into the Cisco Finesse agent desktop. It enables integration of third-party web applications, such as CRMs or internal tools, directly into the Finesse UI.

**Valid 300-835 Dumps** shared by ExamDiscuss.com for Helping Passing 300-835 Exam! ExamDiscuss.com now offer the **newest 300-835 exam dumps**, the ExamDiscuss.com 300-835 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com 300-835 dumps with Test Engine here:

<https://www.examdiscuss.com/Cisco/exam/300-835/premium/> (132 Q&As Dumps, **35%OFF**)

Special Discount Code: **freecram**)

**Valid 300-835 Dumps** shared by ExamDiscuss.com for Helping Passing 300-835 Exam!  
ExamDiscuss.com now offer the **newest 300-835 exam dumps**, the ExamDiscuss.com  
300-835 exam **questions have been updated** and **answers have been corrected** get the  
**newest** ExamDiscuss.com 300-835 dumps with Test Engine here:

<https://www.examdiscuss.com/Cisco/exam/300-835/premium/> (132 Q&As Dumps, **35%OFF**)

Special Discount Code: **freecram**)