

➤ **Vendor:** Cisco

➤ **Exam Code:** 300-835

➤ **Exam Name:** Implementing Automation for Cisco Collaboration Solutions

➤ **New Updated Questions from** [Braindump2go](#) (Updated in [Sep/2020](#))

[Visit Braindump2go and Download Full Version 300-835 Exam Dumps](#)

QUESTION 49

Drag and Drop Question

Drag and drop the XML snippets onto the AXL request to build a request that sets the cluster-wide Call Park Reversion Timer to 2 minutes. Not all options are used.

Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.cisco.com/AXL/API/11.5">
  <soapenv:Header/>
  <soapenv:Body>
    [ ]
    [ ]
    [ ]
    <service>Cisco CallManager</service>
    <value>120</value>
    [ ]
  </soapenv:Body>
</soapenv:Envelope>
```

<name>CallParkDisplayTimer</name>

<name>CallParkReversionTimeout</name>

<ns:updateProcessNodeService>

<processNodeName>Default</processNodeName>

</ns:updateServiceParameter>

<ns:updateServiceParameter>

</ns:updateProcessNodeService>

<processNodeName>EnterpriseWideData</processNodeName>

Answer:

Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.cisco.com/AXL/API/11.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:updateServiceParameter>
      <processNodeName>EnterpriseWideData</processNodeName>
      <name>CallParkReversionTimeout</name>
      <service>Cisco CallManager</service>
      <value>120</value>
    </ns:updateServiceParameter>
  </soapenv:Body>
</soapenv:Envelope>
```

<name>CallParkDisplayTimer</name>

<ns:updateProcessNodeService>

<processNodeName>Default</processNodeName>

</ns:updateProcessNodeService>

QUESTION 50

Drag and Drop Question

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

Answer Area

```
Content-Type: text.xml
Content-Length: 726
Host: cucm.cisco.com:8443
Authorization:
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="
    <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

Basic YWRtaW46cGFzc3dvcmQ=

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

http://www.cisco.com/AXL/API/12.5

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

Bearer YWRtaW46cGFzc3dvcmQ=

SOAPAction: "CUCM:DB ver=12.5 addUcService"

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

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 51

Drag and Drop Question

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

HTTPDigestAuth

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)

```

AXLSoap.xsd

HTTPDigestAuth

createPartition

/axlapi

QUESTION 52

Drag and Drop Question

Drag and drop the code to add a Webex Teams Space Widget in an existing HTML page. Not all items are used.

Answer Area

```
<body>
  <div id="my-webexteams-widget" style="width: 500px; height: 500px;" />
  <script>
    <var widgetEl = document.getElementById(' ');
    // Init a new widget
    ciscopark.widget(widgetEl).  (
      accessToken: 'ACCESS_TOKEN',
      destinationId: 'SPACE_ID',
      destinationType: ' ',
    );
  </script>
</body>
```

my-webexteams-widget

spaceId

mySpaceWidget

webexteams-widget

space_ID

spaceWidget

RoomId

PersonEmail

Answer:

Answer Area

```
<body>
  <div id="my-webexteams-widget" style="width: 500px; height: 500px;" />
  <script>
    <var widgetEl = document.getElementById(' my-webexteams-widget ');
    // Init a new widget
    ciscopark.widget(widgetEl). spaceWidget (
      accessToken: 'ACCESS_TOKEN',
      destinationId: 'SPACE_ID',
      destinationType: ' spaceId ',
    );
  </script>
</body>
```

mySpaceWidget

webexteams-widget

space_ID

RoomId

PersonEmail

QUESTION 53

Drag and Drop Question

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.



Python Yesterday, 11:13

The bot says:

I am a **developer** tool!

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** tool!"

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 = { "Authentication" : "Bearer" % 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.ciscospark.com/v1/messages",
                        headers=header, data= json.dumps(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...")
```

"PUT"	payload
"Authorization"	status
"Bearer %s"	code
json(payload)	

QUESTION 54

Drag and Drop Question

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

<pre>payload = { 'roomId: resp1.json()['id'], 'text': 'test' } resp2 = requests.post(url + '/messages', headers=headers, json=payload)</pre>
<pre>if resp2.status_code == 200: print('Success!')</pre>
<pre>headers = { 'Content-type': 'application/json', 'Authorization': 'Bearer YTQzMzBiZmItNDJkNCO643-417974-ad72cae0e10f' } url = 'https://api.ciscospark.com/v1'</pre>
<pre>payload = { 'title': 'test' } resp1 = requests.post(url + '/rooms', headers=headers, json=payload)</pre>
<pre>payload = { 'roomId: resp1.json()['id'], 'personEmail': 'test@test.com' } requests.post(url + '/memberships', headers=headers, json=payload)</pre>

Answer:

Answer Area

```
headers = { 'Content-type': 'application/json',
'Authorization': 'Bearer YtQzMzBiZmItNDJkNC0643-417974-ad72cae0e10f' }
url = 'https://api.ciscospark.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!')
```

QUESTION 55

Drag and Drop Question

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()
```

response['id']

personEmail

json

data

Authorization

membershipEmail

apiKey

Authentication

OAuth

response.id

response[body]['id']

Answer:

Answer Area

```
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']

json

membershipEmail

apiKey

Authentication

OAuth

response[body]['id']

QUESTION 56

Drag and Drop Question

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.

Answer Area

```
import requests

head = { 'Content-Type': ' ',
        'Authorization': 'Bearer NWU4NjQ0YWU0tNjYItMWMY_-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

Answer:

Answer Area

```
import requests

head = { 'Content-Type': ' application/json ',
        'Authorization': 'Bearer NWU4NjQ0YWU0tNjYItMWMY_-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 57

Drag and Drop Question

Drag and drop the code to create a Webex Teams Recents Widget in an existing HTML page. Not all items are used.

Answer Area

```
<div style="width:300px; height: 500px;"
  id="recents"
/>
<script>
  // Initialize a new widget

  const widgetElement = document.getElementById(' ');

  ciscospark.widget(widgetElement).({
    accessToken: "YOUR_WEBEX_TEAMS_ACCESS_TOKEN",
    onEvent: callback
  });

  function callback(type,event) {

    if (type == ' ') {
      console.log("room " + event.data.id + " was selected");
    }
  }
</script>
```

recentsId

selected

my-recents-widget

rooms:selected

recentsWidget

recents

Answer:**Answer Area**

```
<div style="width:300px; height: 500px;"
  id="recents"
/>
<script>
  // Initialize a new widget

  const widgetElement = document.getElementById('my-recents-widget');

  ciscospark.widget(widgetElement).recentsWidget({
    accessToken: "YOUR_WEBEX_TEAMS_ACCESS_TOKEN",
    onEvent: callback
  });

  function callback(type,event) {

    if (type == 'recents') {
      console.log("room " + event.data.id + " was selected");
    }
  }
</script>
```

recentsId

selected

rooms:selected

QUESTION 58

Drag and Drop Question

Refer to the exhibit. 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.


```
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=Y21zY29zcGFyazovMGNmNi0xMWVhLTk1ODktNzMyYjA4Zj1hYzE1",
  "orgId": "Y21zY29zcGFyazovL3VzL09kZi05NjQzLTQxN2YtOTk3NC1hZDcyY2F1MGUxMGY",
  "createdBy": "Y21zY29zcGFyazovL3VzL1l1ZTdiLTRkNjktOGYzOS0wNGI2YTFlNzE1ZmU",
  "appId": "Y21zY29zcGFy5YmElMmZlY2JiMmM3ZDUxNWNiNGEwY2M5MWFh",
  "ownedBy": "creator",
  "status": "active",
  "created": "2019-11-22T07:08:51.934Z",
  "actorId": "Y21zY29zcGFyazovL3VzL1dFQkhPT0svYmI5MGE5MDAtNjM4YyYzRiOThhMjk5",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzL01FU1mNy0xMWVhLTk1ODktNzMyYjA4Zj1hYzE1",
    "roomId": "Y21zY29zcGFyazovMGNmNi0xMWVhLTk1ODktNzMyYjA4Zj1hYzE1",
    "roomType": "group",
    "personId": "Y21zY29zcGFyazovhNDEtOGM0MS0yYWJkZjg5ZjQ0ZjQ",
    "personEmail": "user@example.com",
    "mentionedPeople": [
      "Y21zY29zcGFyazovL3VzLjktOGYzOS0wNGI2YTFlNzE1ZmU"
    ]
  },
  "created": "2019-11-22T07:09:20.7002"
}
```

Answer Area

```
curl -X [ ] https://api.ciscospark.com/v1/ [ ] \
-H ' [ ] 'NTIwNTUyOWtNDkz_Pf84-9974-ad72ce10f' \
-H 'Content-Type: application/json' \
-d '{
  "name": "exam question",
  "resource": "messages",
  "event": "created",
  "filter": "roomId= Y21zY29zcGFyazovMGNmNi0xMWVhLTk1ODktNzMyYjA4Zj1hYzE1",
  "targetUrl": "https://2615c331.ngrok.io",
  [ ]
}'
```

"secret": "VERY SECRET"	"actor":	Authentication: Bearer
Authentication: OAuth	GET	webhooks
POST	"status": "active"	messages
events	Authorization: Bearer	PUT
Authorization: OAuth	"actor":	

Answer:

Answer Area

```
curl -X [ ] POST https://api.ciscospark.com/v1/ [ ] messages \
-H ' [ ] Authentication: Bearer 'NTIwNTUyOWtNDkz_Pf84-9974-ad72ce10f' \
-H 'Content-Type: application/json' \
-d '{
  "name": "exam question",
  "resource": "messages",
  "event": "created",
  "filter": "roomId= Y21zY29zcGFyazovMGNmNi0xMWVhLTk1ODktNzMyYjA4Zj1hYzE1",
  "targetUrl": "https://2615c331.ngrok.io",
  [ ] "status": "active"
}'
```

"secret": "VERY SECRET"	"actor":	
Authentication: OAuth	GET	webhooks
events	Authorization: Bearer	PUT
Authorization: OAuth	"actor":	

QUESTION 59

Drag and Drop Question

Drag and drop the commands from the left onto the curl command to create a custom branding message using the xAPI HTTP API. Not all options are used.

	Answer Area
POST	<pre>curl -X [] \ http://192.168.1.34/[] \ -H 'Authorization: Basic aW50ZWdyYXRvcjppbnRlZ3JhdG9yDQo=' \ -H '[]' \ -d '[]' <UserInterface> <CustomMessage>My Company Branding</CustomMessage> </UserInterface> []'</pre>
Content-Type: text/xml	
PUT	
putxml	
<BrandingConfig>	
</BrandingConfig>	
<Branding>	
postxml	
Content: text	
</Branding>	
<Configuration>	
</Configuration>	

Answer:

	Answer Area
PUT	<pre>curl -X POST [] \ http://192.168.1.34/[] \ -H 'Authorization: Basic aW50ZWdyYXRvcjppbnRlZ3JhdG9yDQo=' \ -H 'Content-Type: text/xml' \ -d '<Configuration> <UserInterface> <CustomMessage>My Company Branding</CustomMessage> </UserInterface> </Configuration>'</pre>
<BrandingConfig>	
</BrandingConfig>	
<Branding>	
postxml	
Content: text	
</Branding>	