

➤ **Vendor:** Cisco➤ **Exam Code:** 300-835➤ **Exam Name:** Implementing Automation for Cisco Collaboration Solutions➤ **New Updated Questions from** [Braindump2go](#) (Updated in [May/2020](#))**[Visit Braindump2go and Download Full Version 300-835 Exam Dumps](#)****QUESTION 23**
DRAG DROP

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

Select and Place:**Answer Area**

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

my-webexteams-widget	spaceId	mySpaceWidget
webexteams-widget	space ID	spaceWidget
RoomId	PersonEmail	

Correct 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
    ciscooark.widget(widgetEl), {
      spaceWidget
      accessToken: 'ACCESS_TOKEN',
      destinationId: 'SPACE_ID',
      destinationType: ' spaceId ',
    });
  </script>
</body>
```

my-webexteams-widget	spaceId	mySpaceWidget
webexteams-widget	space ID	spaceWidget
RoomId	PersonEmail	

QUESTION 24
DRAG DROP

	Python Yesterday, 11:13
Bot	The bot says:
	I am a developer too!

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.

Select and Place:[300-835 Exam Dumps](#) [300-835 Exam Questions](#) [300-835 PDF Dumps](#) [300-835 VCE Dumps](#)<https://www.braindump2go.com/300-835.html>

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** lol!"

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

Correct Answer:

Answer Area

```

import requests, json, os

header = { "Authorization" : "Bearer" & os.environ.get("BOT_TOKEN"),
           "Content-Type": "application/json"}

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

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.[ ] == 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

QUESTION 25

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

QUESTION 26

DRAG DROP

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.

Select and Place:

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 YTO2MzBiZmItNDJkNCO643-417974-ad720ac0c10f' }
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)
```

Correct Answer:

Answer Area

```
headers = { 'Content-type': 'application/json',
            'Authorization': 'Bearer YTO2MzBiZmItNDJkNCO643-417974-ad720ac0c10f' }
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!')
```

```
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 YTO2MzBiZmItNDJkNCO643-417974-ad720ac0c10f' }
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)
```

QUESTION 27

When a Webex site has single-sign-on enabled, which type of data is required in the body element of an AuthenticateUser XML API Request?

- A. Webex username and password
- B. Webex Teams OAuth access token
- C. JWT token
- D. BOT access token

Correct Answer: B

QUESTION 28

DRAG DROP

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.

Select and Place:

[300-835 Exam Dumps](#) [300-835 Exam Questions](#) [300-835 PDF Dumps](#) [300-835 VCE Dumps](#)

<https://www.braindump2go.com/300-835.html>

Answer Area

```

import requests

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

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

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

Correct 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, [redacted] data body)
    return response

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

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

QUESTION 29

```

import requests

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

Refer to the exhibit. 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`

[0-835 Exam Questions](#) [300-835 PDF Dumps](#) [300-835 VCE Dumps](#)

<https://www.braindump2go.com/300-835.html>



Correct Answer: C

QUESTION 30

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:

```

import requests

head = { 'Content-Type': 'application/json',
        'Authorization': 'Bearer NWU4NjQ0YWU0NjY1tWMMY_417f_ae0ae10f' }

res = requests.post(url = 'https://api.ciscospark.com/v1/rooms/',
                   headers = head, json = { 'populateMembers': True })

spaceId = res.json()[0]['id']

members = ['ToEnglish@webex.bot', 'ToSpanish@webex.bot']
for member in members:
    requests.post(url='https://api.ciscospark.com/v1/rooms/{}/members'.format(spaceId),
                  headers = head,
                  json = { 'roomId': spaceId, 'member': member })

```

Correct Answer:

Answer Area

```
import requests

head = { 'Content-Type': 'application/json ',
        'Authorization': 'Bearer NWU4NjQ0YVWUitNjYitMWMY_417f ae0ae10f' }

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})
```

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

QUESTION 31

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 32

DRAG DROP

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

Select and Place:

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

Correct Answer:



```
<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>
```

recents

```
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": "Y21zY29zcGFyY5YmElMmZlY2JiMmM3ZDUxNWNIiGEWY2M5MWFh",
  "ownedBy": "creator",
  "status": "active",
  "created": "2019-11-22T07:08:51.934Z",
  "actorId": "Y21zY29zcGFyazovL3VLRhNDEtOGM0MS0yYWJkZjg5ZjQ0ZjQ",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzL01FU1mNy0xMWVhLTk1ODktNzMyYjA4ZjlhYze1",
    "roomId": "Y21zY29zcGFyazovMGNmNi0xMWVhLTk1ODktNzMyYjA4ZjlhYze1",
    "roomType": "group",
    "personId": "Y21zY29zcGFyazovhNDEtOGM0MS0yYWJkZjg5ZjQ0ZjQ",
    "personEmail": "user@example.com",
    "mentionedPeople": [
      "Y21zY29zcGFyazovL3VzLjktOGYzOS0wNGI2YIFlNzE1ZmU"
    ],
    "created": "2019-11-22T07:09:20.7002"
  }
}
```

Select and Place:

Answer Area

```
curl -X [ ] https://api.ciscospark.com/v1/ [ ]
-H ' [ ] "NT1wNTHy2WLN0kx IP84-9974-ad72ad107"
-H 'Content-Type: application/json'
-d '{
  "name": "exam question",
  "resource": "messages",
  "event": "created",
  "filter": "roomId= Y2lzy29zcGFyazovMGNmN10xMWVhLlk1ODktdmYyJA4zjlhYzE1",
  "targetUrl": "https://2615c331.ngrok.io",
  [ ]
}
```

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

Correct Answer: