Free Questions for 300-835 by ebraindumps

Shared by Conley on 29-01-2024

For More Free Questions and Preparation Resources

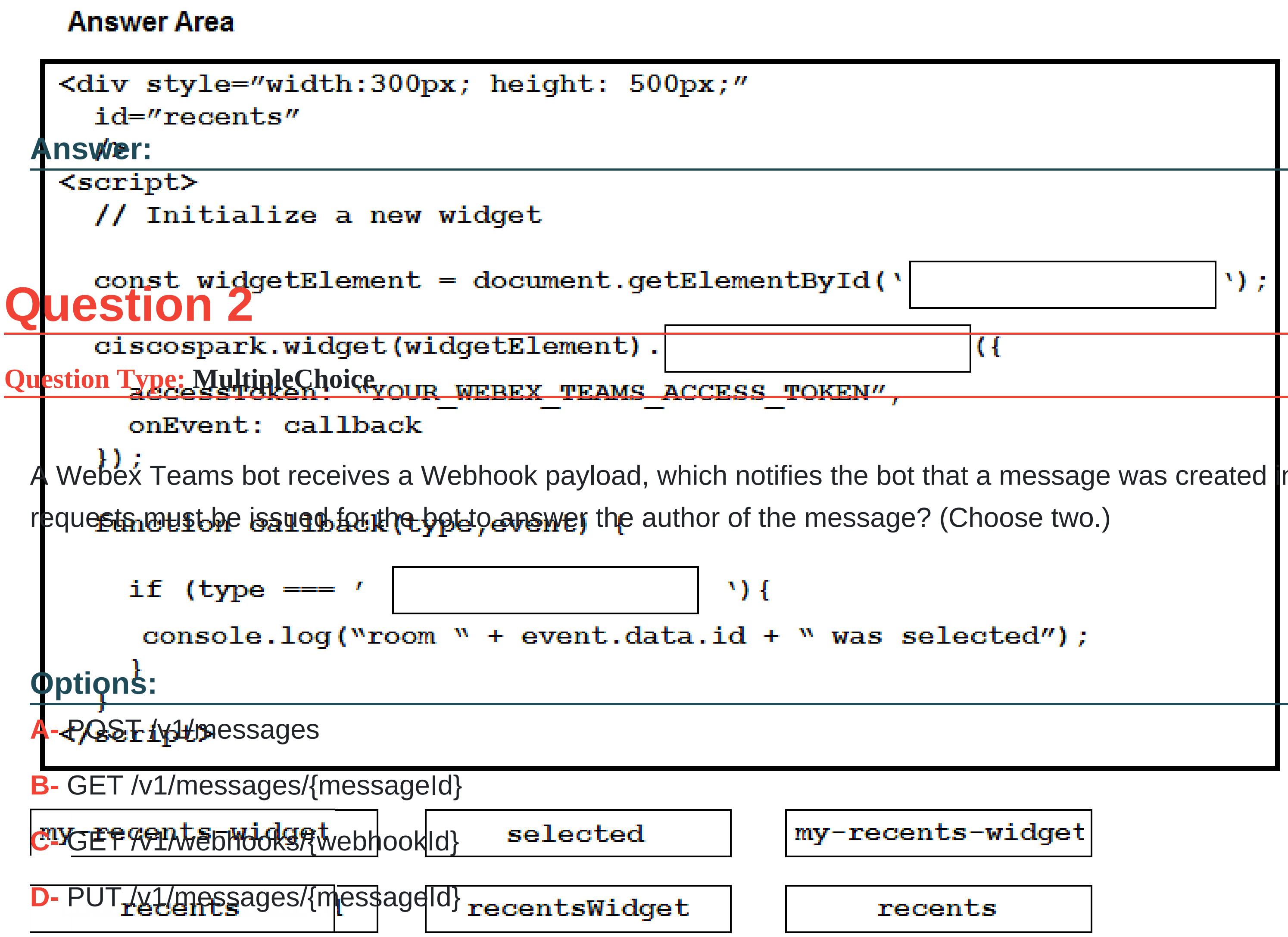
Check the Links on Last Page

Question 1

Question Type: DragDrop

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

Answer:



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>
```

my-recents-widget

selected

my-recents-widget

recents

recentsWidget

recents

Question 2

Question Type: MultipleChoice

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.)

Options:

A- POST /v1/messages

B- GET /v1/messages/{messageId}

C- GET /v1/webhooks/{webhookId}

D- PUT /v1/messages/{messageId}

E- POST /v1/webhooks

Answer:

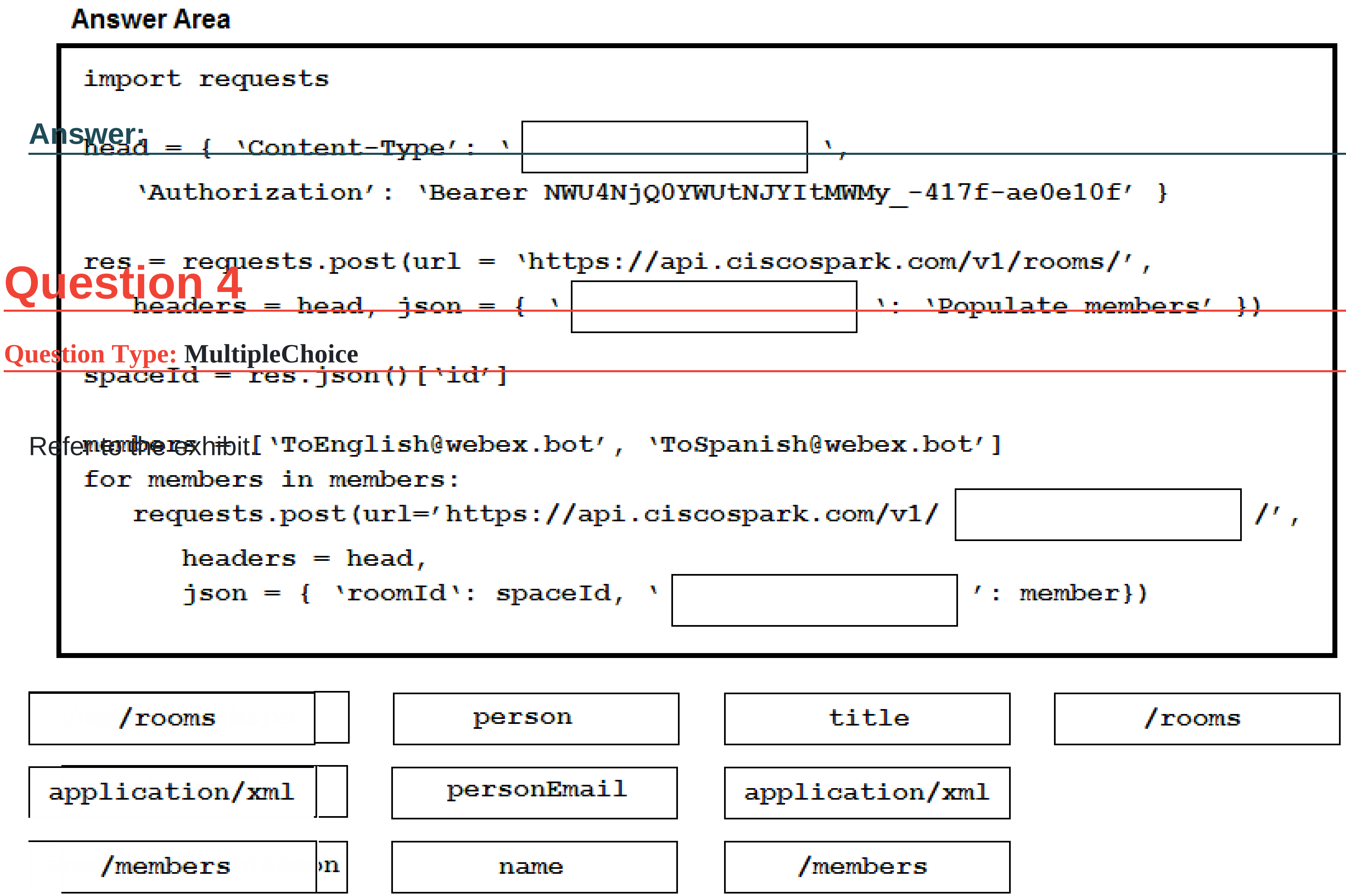
B, E

Question 3

Question Type: DragDrop

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:



Answer Area

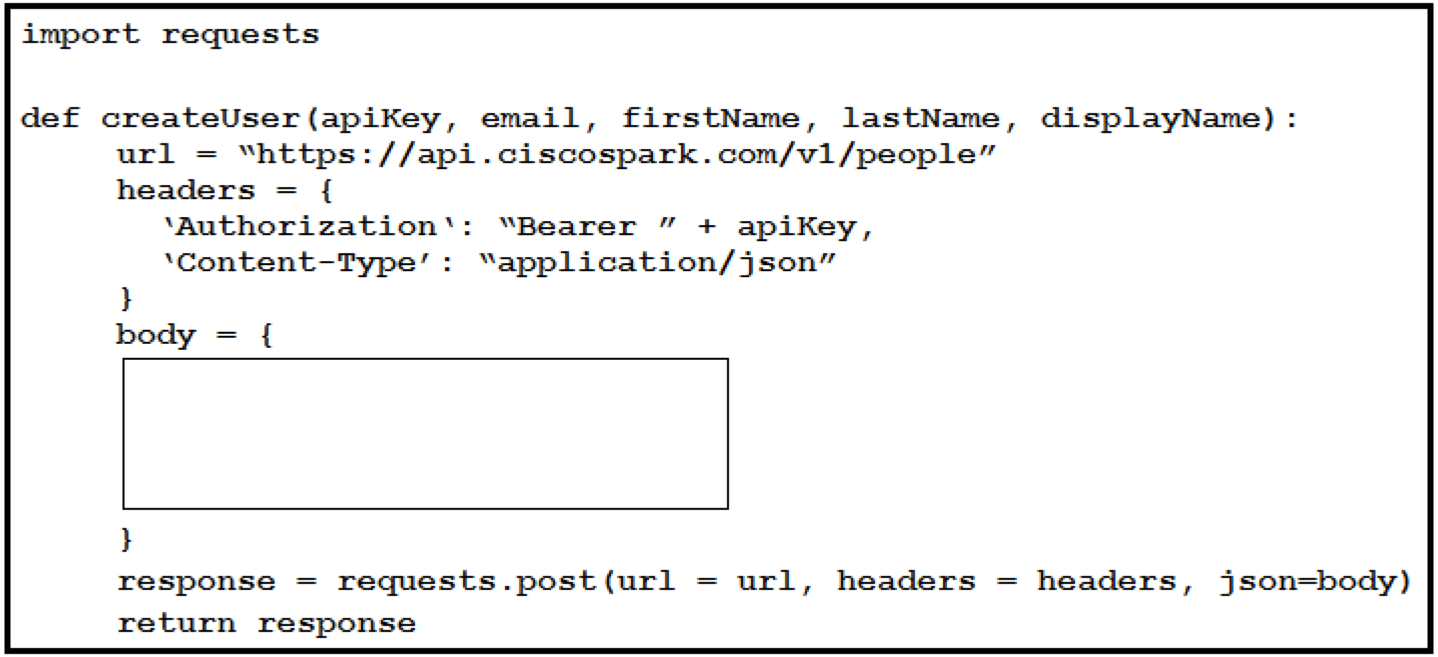
```
import requests

head = { ‘Content-Type’: ‘                    ’,
    ‘Authorization’: ‘Bearer NWU4NjQ0YWUtNJYItMWMy_-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})
```

/rooms	person	title	/rooms
application/xml	personEmail	application/xml	
/members	name	/members	

Question 4

Question Type: MultipleChoice

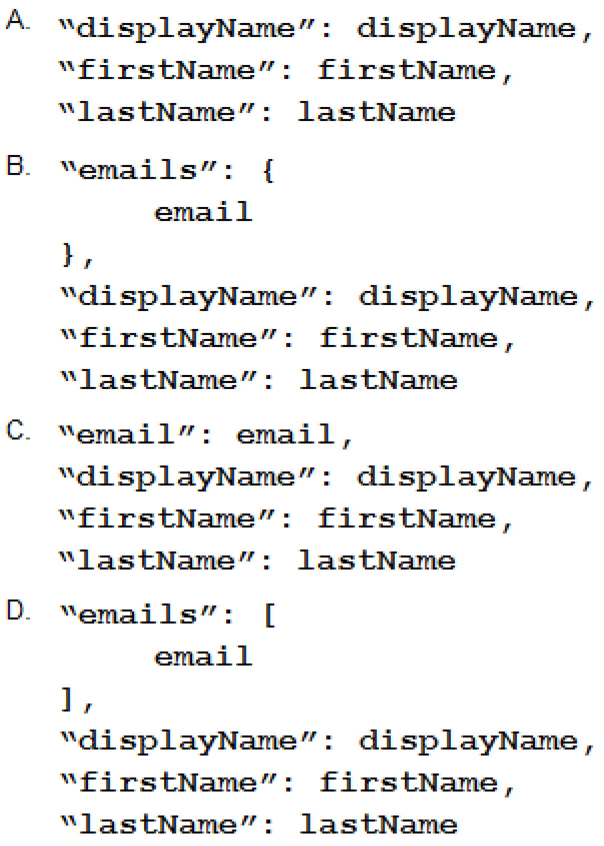
Refer to the exhibit.



```
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 = {



     }
     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
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

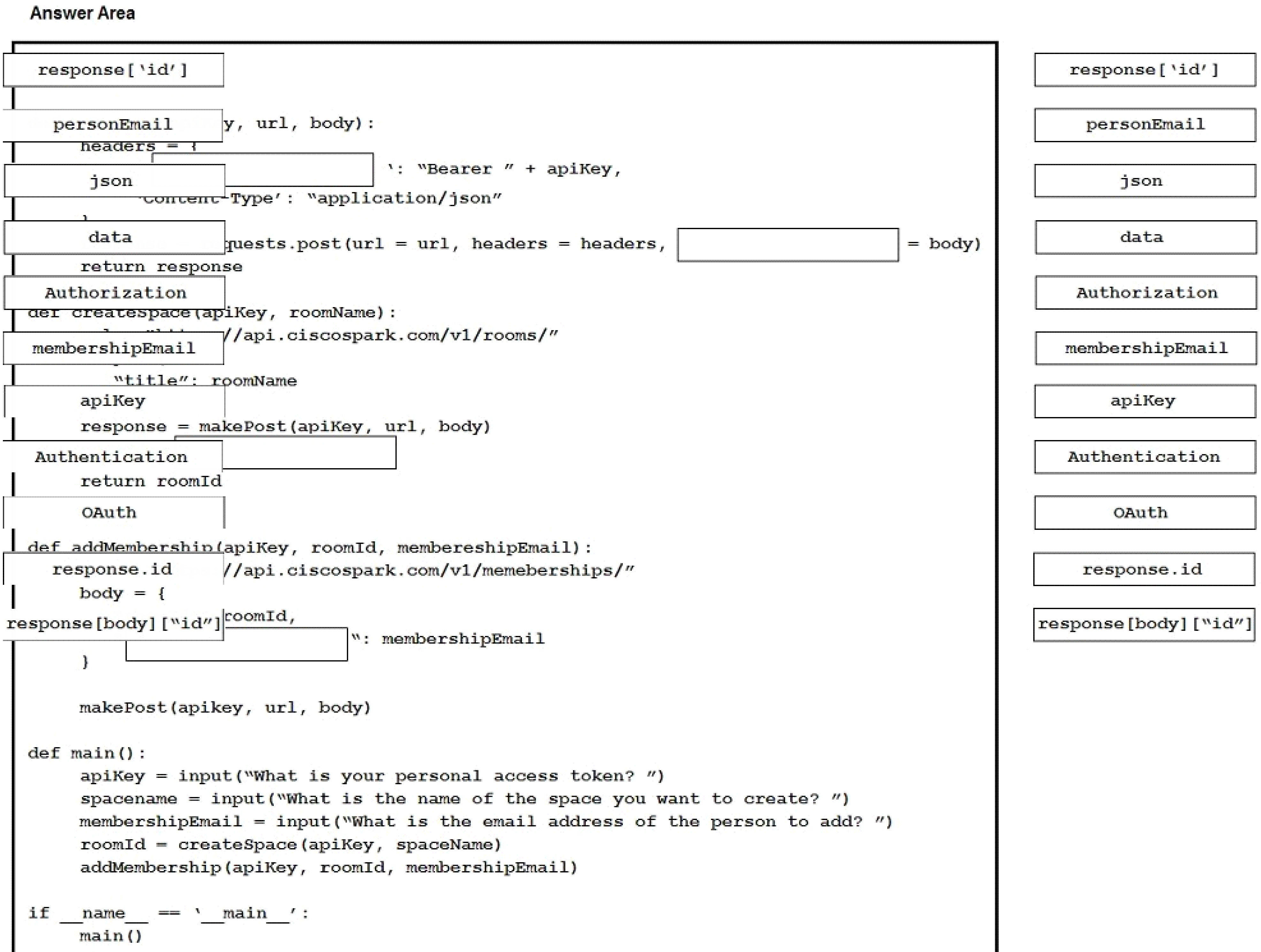
Answer:

D

Question 5

Question Type: **DragDrop**

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

```
                   y, url, body):
    headers = {
        '[          ]': "Bearer " + apiKey,
        'Content-Type': "application/json"
    }
    quests.post(url = url, headers = headers, [          ] = body)
    return response

def createSpace(apiKey, roomName):
               //api.ciscospark.com/v1/rooms/"

        "title": roomName

    response = makePost(apiKey, url, body)
    [          ]
    return roomId

def addMembership(apiKey, roomId, membereshipEmail):
               //api.ciscospark.com/v1/memeberships/"
    body = {
               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"]

response['id']
personEmail
json
data
Authorization
membershipEmail
apiKey
Authentication
OAuth
response.id
response[body]["id"]

Answer:

Question 6

Question Type: MultipleChoice

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?

Options:

A- Webex username and password

B- Webex Teams OAuth access token

C- JWT token

D- BOT access token

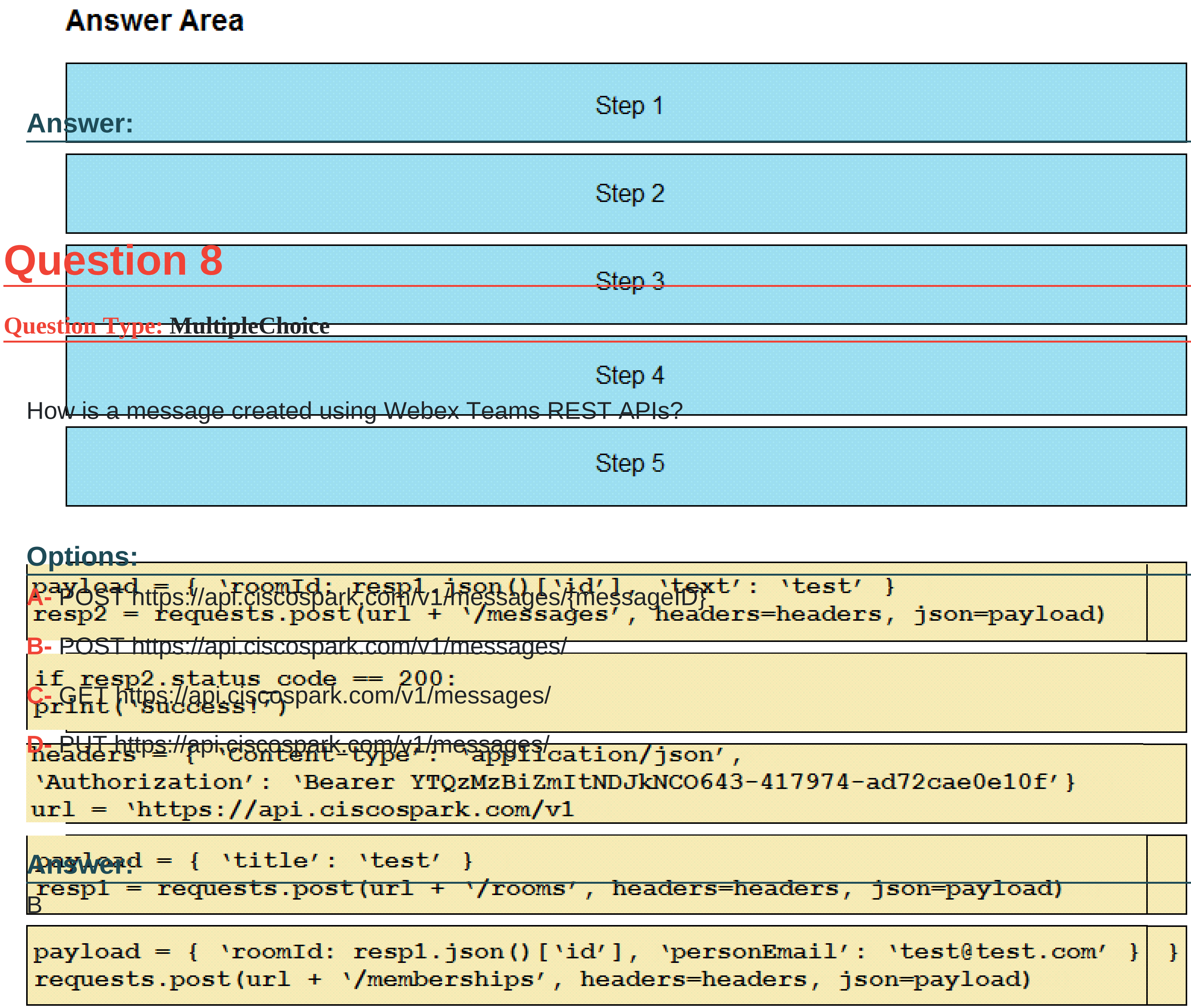
Answer:

B

Question 7

Question Type: **DragDrop**

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 YTQzMzBiZmItNDJkNC0643-417974-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:

Question 8

Question Type: **MultipleChoice**

How is a message created using Webex Teams REST APIs?

Options:

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/

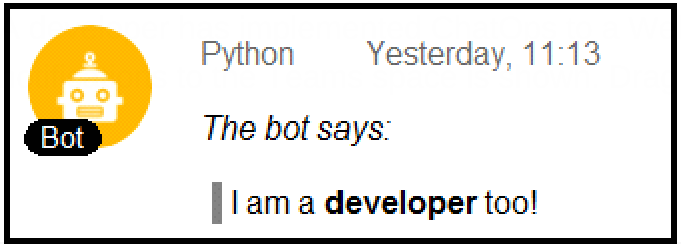
Answer:

B

Question 9

Question Type: DragDrop

Refer to the exhibit.



Python Yesterday, 11:13

The bot says:

> I am a **developer** too!

bex Teams space as described in the exhibit. The Python script that pushes the and drop the code to complete the script. Not all options are used.

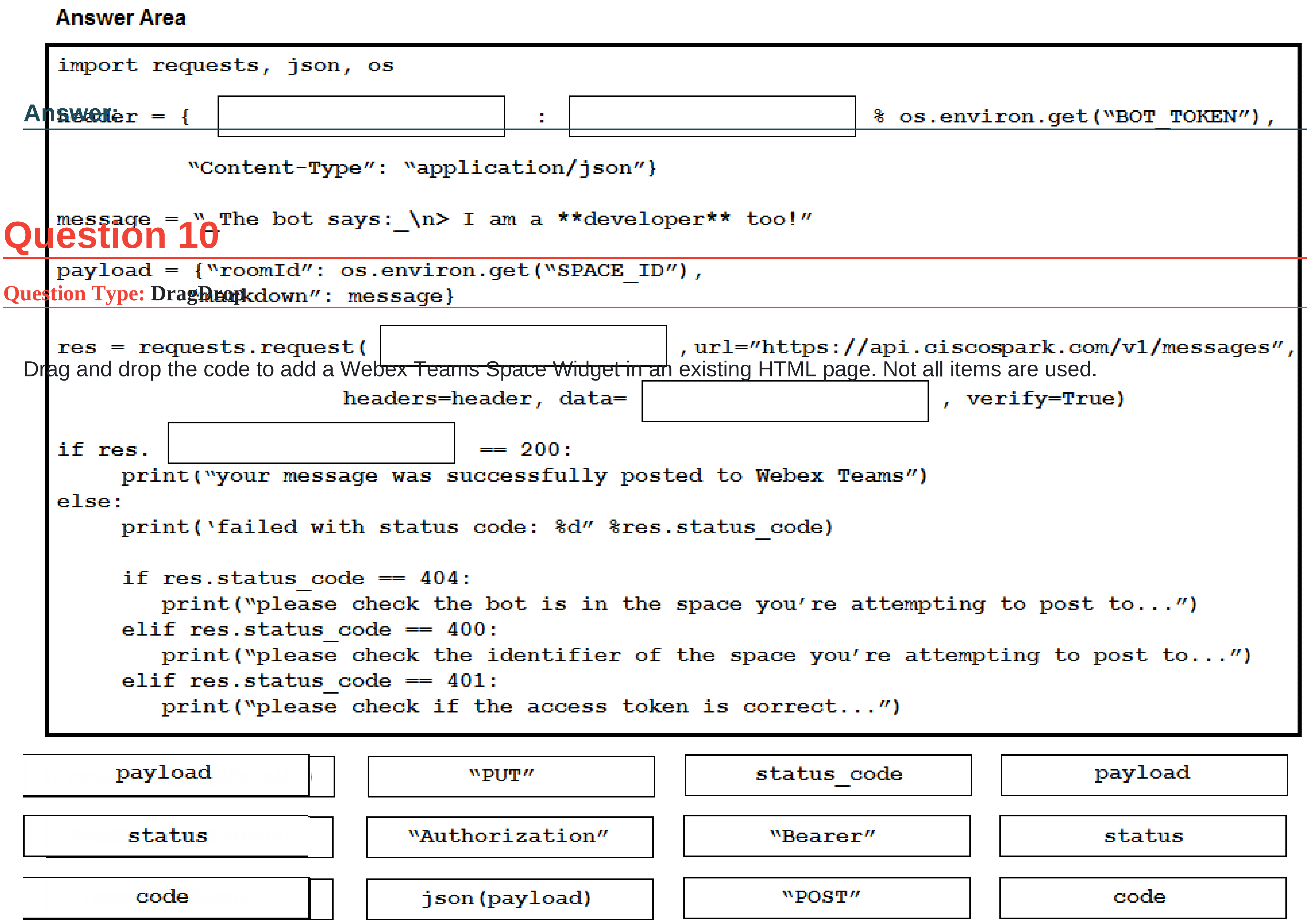
`payload`

Answer:

Question 10

Question Type: DragDrop

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

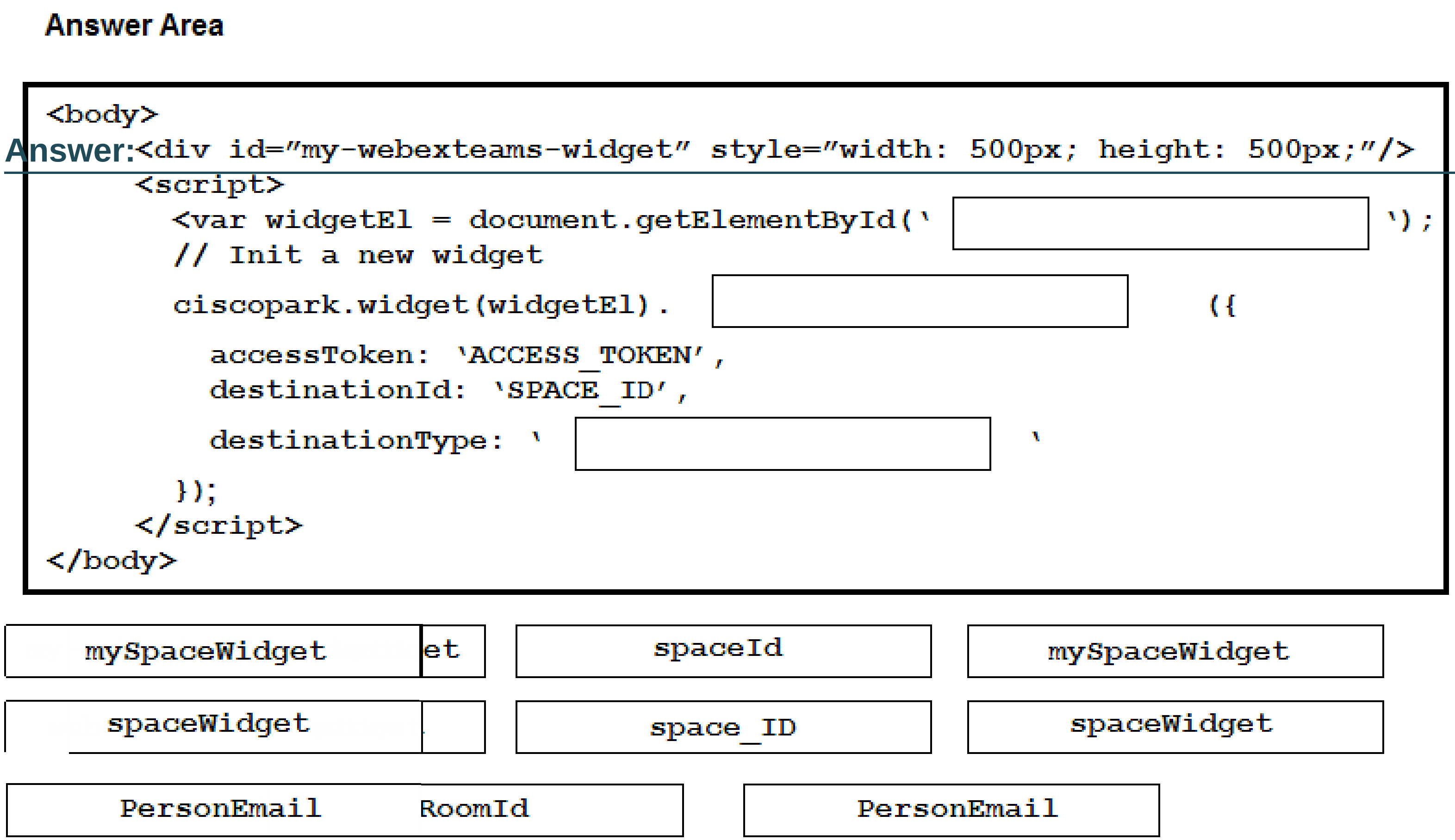


Answer Area

```
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...")
```

payload	"PUT"	status_code	payload
status	"Authorization"	"Bearer"	status
code	json(payload)	"POST"	code

Answer:



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>
```

mySpaceWidget et

spaceId

mySpaceWidget

spaceWidget

space_ID

spaceWidget

PersonEmail RoomId

PersonEmail