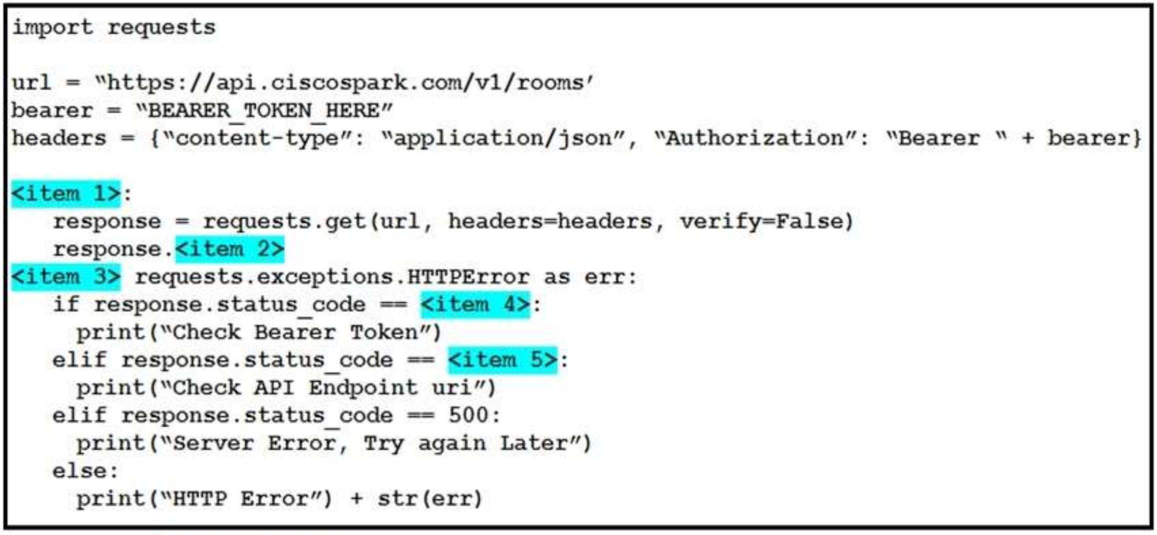
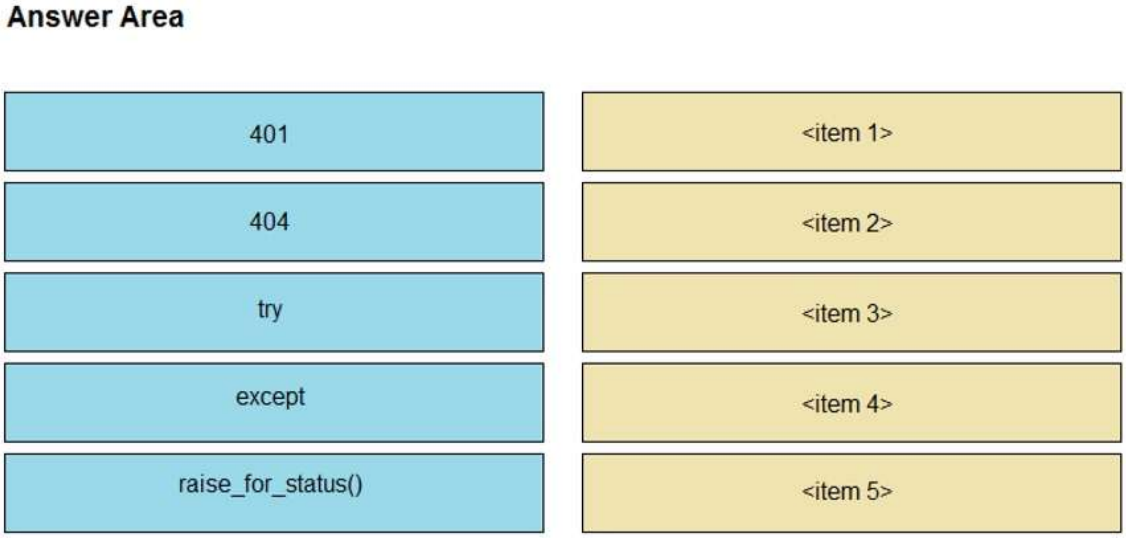
QUESTION 27
Drag and Drop Question
A developer is creating a Python script to use the Webex Teams REST API to list joined spaces, and gracefully handle and print the errors it receives. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the script.



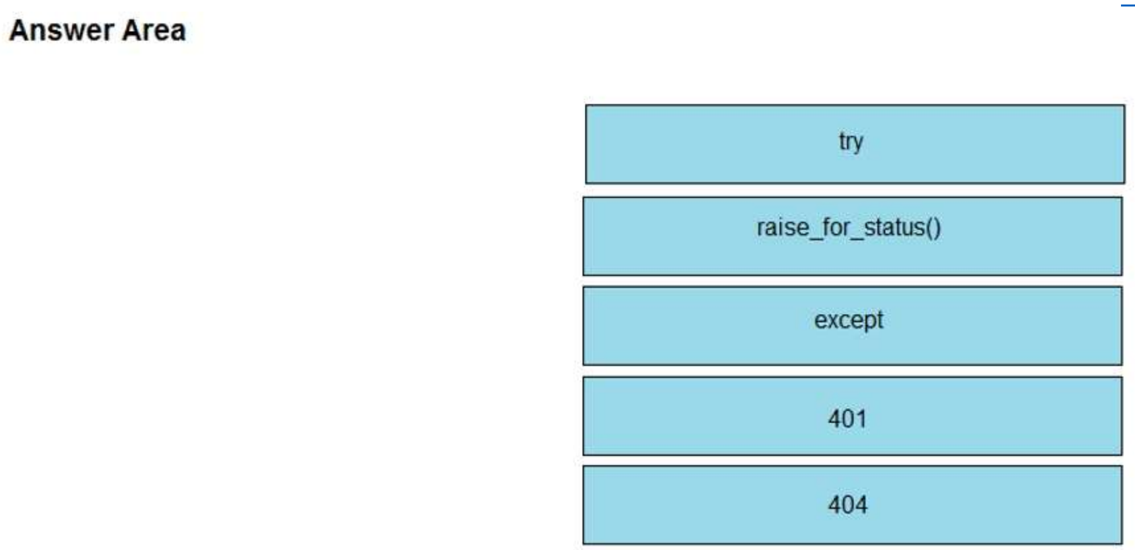
```
import requests

url = “https://api.ciscospark.com/v1/rooms’
bearer = “BEARER_TOKEN_HERE”
headers = {“content-type”: “application/json”, “Authorization”: “Bearer ” + bearer}

<item 1>:
   response = requests.get(url, headers=headers, verify=False)
   response.<item 2>
<item 3> requests.exceptions.HTTPError as err:
   if response.status_code == <item 4>:
     print(“Check Bearer Token”)
   elif response.status_code == <item 5>:
     print(“Check API Endpoint uri”)
   elif response.status_code == 500:
     print(“Server Error, Try again Later”)
   else:
     print(“HTTP Error”) + str(err)
```



Answer Area

401	<item 1>
404	<item 2>
try	<item 3>
except	<item 4>
raise_for_status()	<item 5>

Answer:



Answer Area

try

raise_for_status()

except

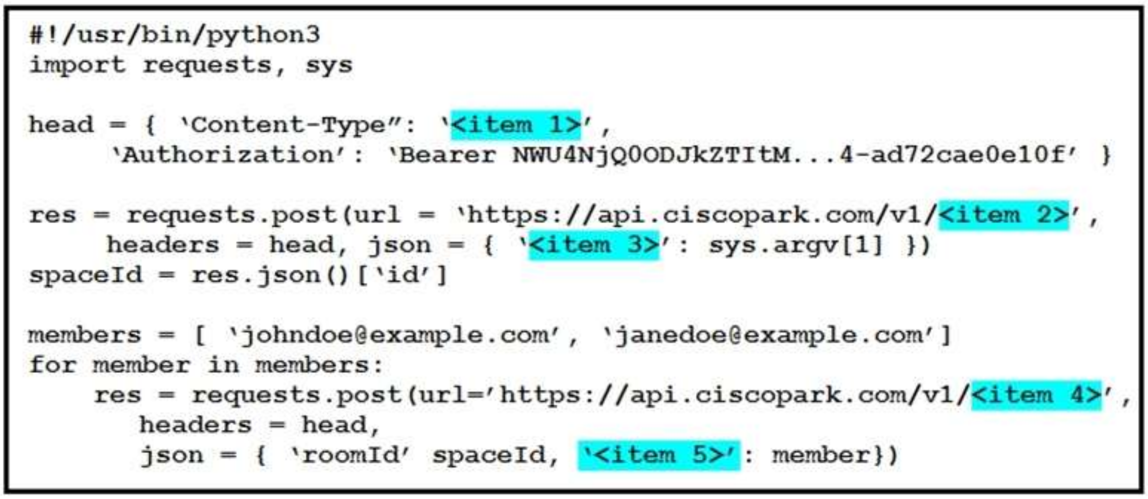
401

404

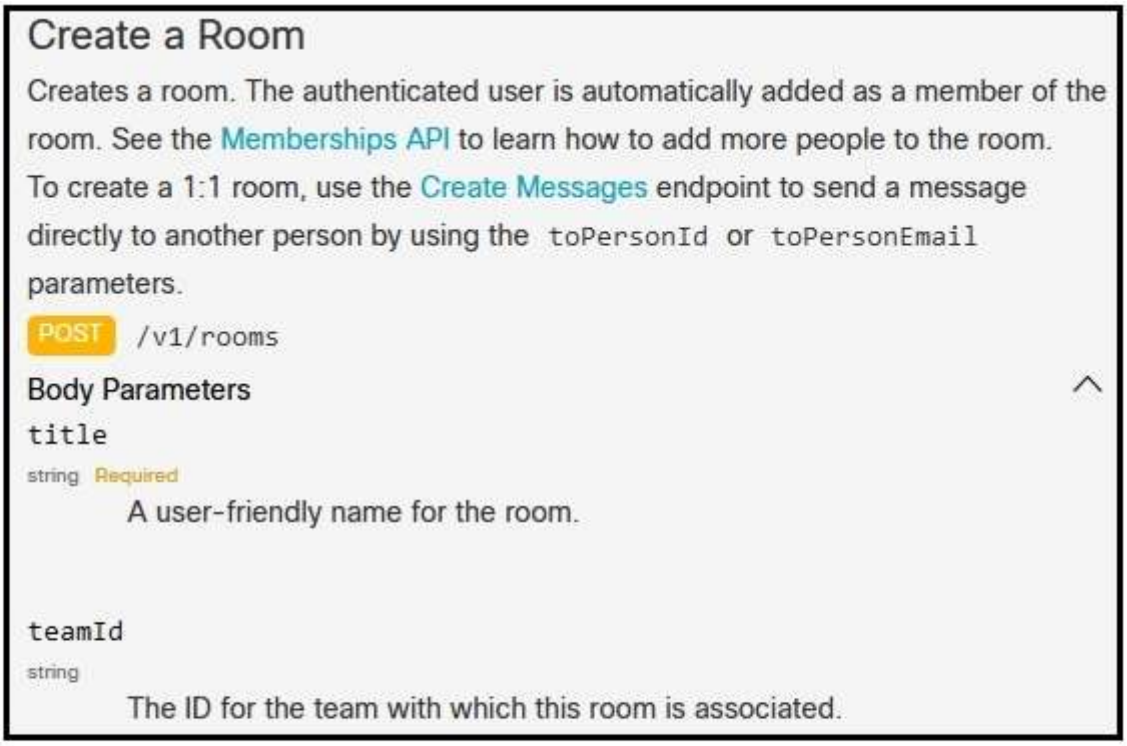
QUESTION 28

Drag and Drop Question

Refer to the exhibit above and click on the resource tabs in the top left corner to view resources to help with this question.



```
#!/usr/bin/python3
import requests, sys

head = { 'Content-Type": '<item 1>',
     'Authorization': 'Bearer NWU4NjQ0ODJkZTItM...4-ad72cae0e10f' }

res = requests.post(url = 'https://api.ciscopark.com/v1/<item 2>',
     headers = head, json = { '<item 3>': sys.argv[1] })
spaceId = res.json()['id']

members = [ 'johndoe@example.com', 'janedoe@example.com']
for member in members:
    res = requests.post(url='https://api.ciscopark.com/v1/<item 4>',
       headers = head,
       json = { 'roomId' spaceId, '<item 5>': member})
```



Create a Room

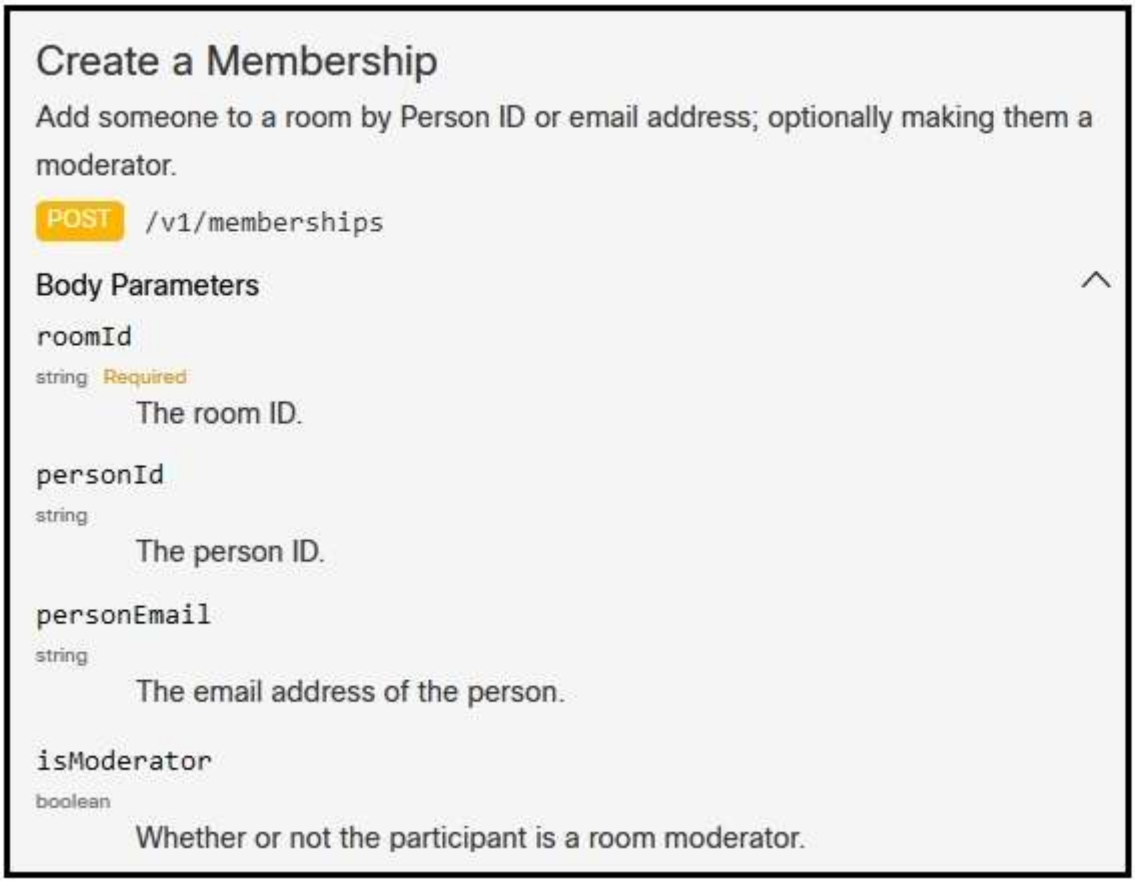
Creates a room. The authenticated user is automatically added as a member of the room. See the Memberships API to learn how to add more people to the room. To create a 1:1 room, use the Create Messages endpoint to send a message directly to another person by using the `toPersonId` or `toPersonEmail` parameters.

`/v1/rooms`

Body Parameters

`title`
string Required
A user-friendly name for the room.

`teamId`
string
The ID for the team with which this room is associated.



Create a Membership

Add someone to a room by Person ID or email address; optionally making them a moderator.

`/v1/memberships`

Body Parameters

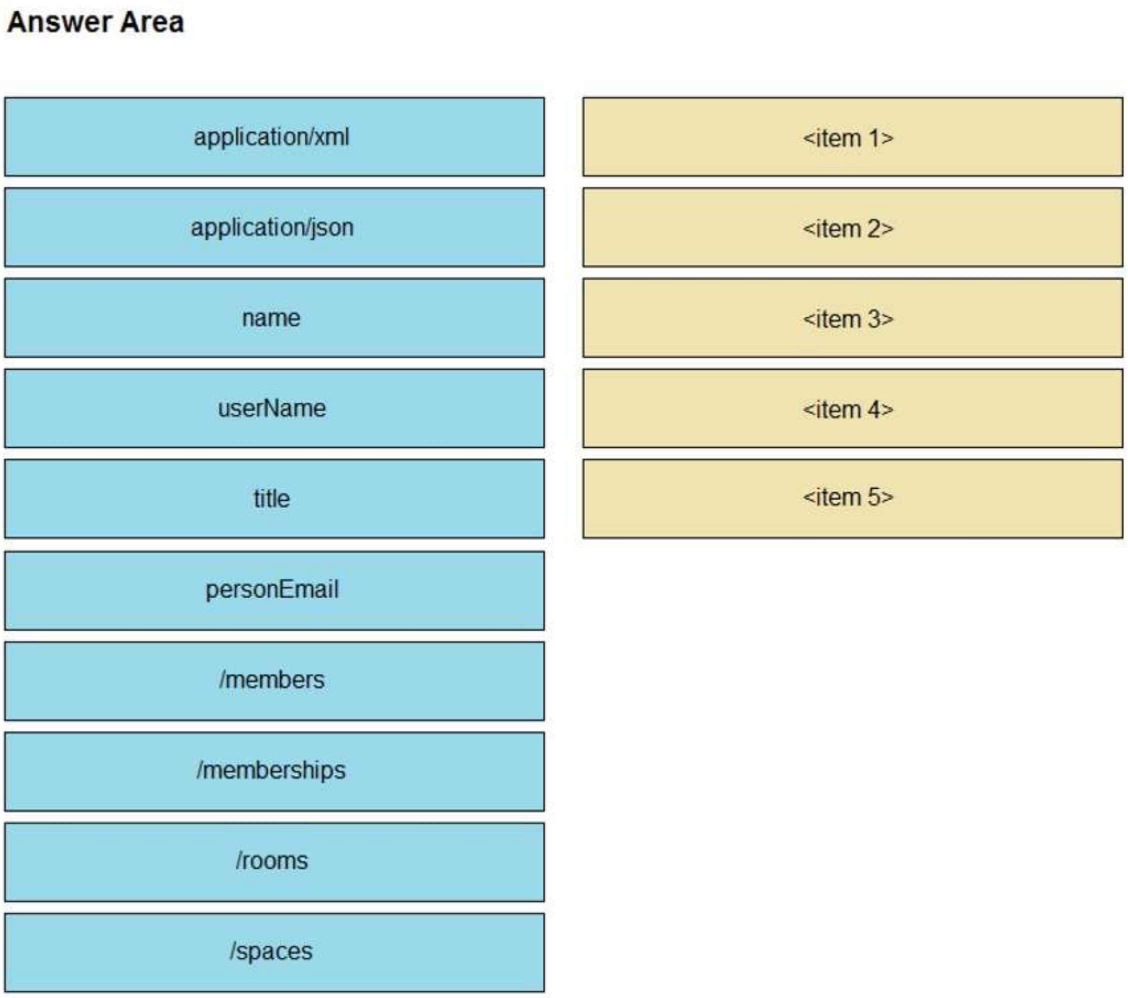
`roomId`
string Required
The room ID.

`personId`
string
The person ID.

`personEmail`
string
The email address of the person.

`isModerator`
boolean
Whether or not the participant is a room moderator.

A developer is creating a Python Script that will use the Webex Teams REST API to automatically create a new collaboration space with him and his team leads on-demand via a Linux terminal command. Drag and drop the code snippets from the left onto the numbers on the right that match the missing sections in the exhibit to complete the script. Not all code snippets are used.



Answer Area

application/xml

application/json

name

userName

title

personEmail

/members

/memberships

/rooms

/spaces

<item 1>

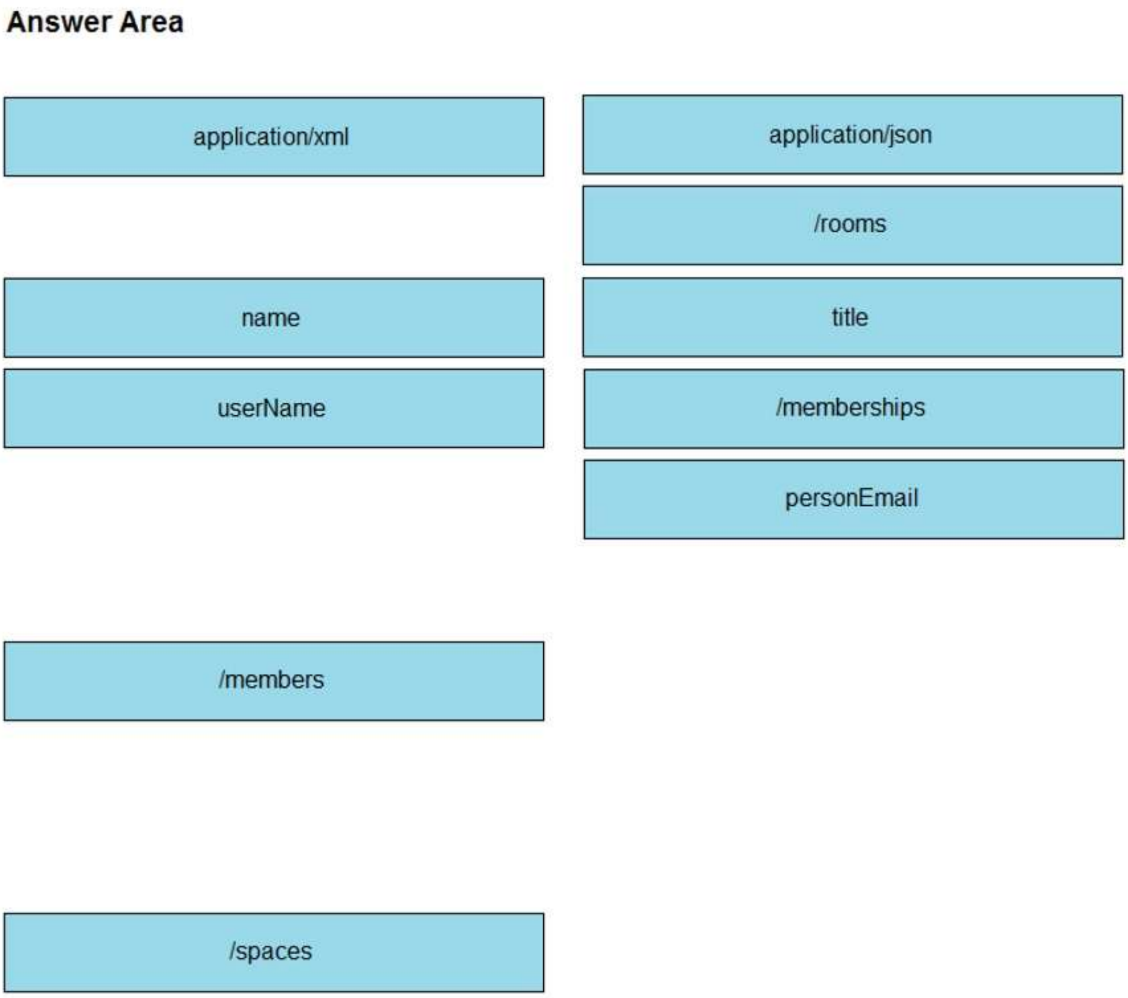
<item 2>

<item 3>

<item 4>

<item 5>

Answer:



Answer Area

application/xml

application/json

/rooms

name

title

userName

/memberships

personEmail

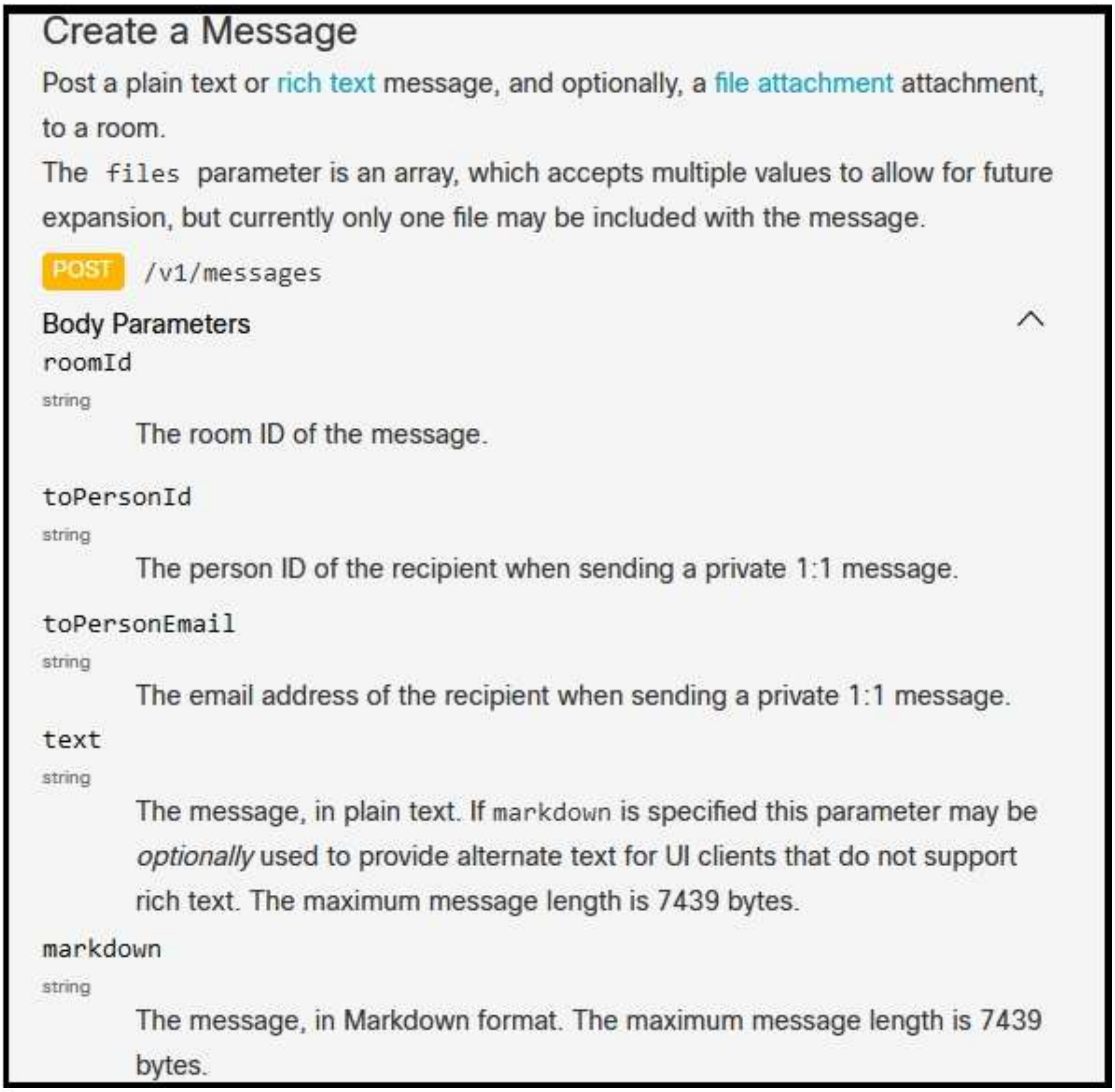
/members

/spaces

QUESTION 29

Drag and Drop Question

Refer to the exhibit. A system administrator has installed a Linux-based alarm system in their home that can execute a Bash shell script when an intruder is detected. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to create a chat-ops script that will notify of alarms via the Webex Teams REST API. Not all code snippets are used.



Create a Message

Post a plain text or rich text message, and optionally, a file attachment attachment, to a room.

The `files` parameter is an array, which accepts multiple values to allow for future expansion, but currently only one file may be included with the message.

POST `/v1/messages`

Body Parameters

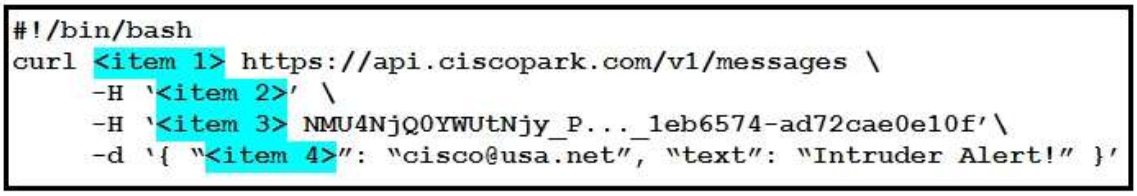
`roomId`
string
The room ID of the message.

`toPersonId`
string
The person ID of the recipient when sending a private 1:1 message.

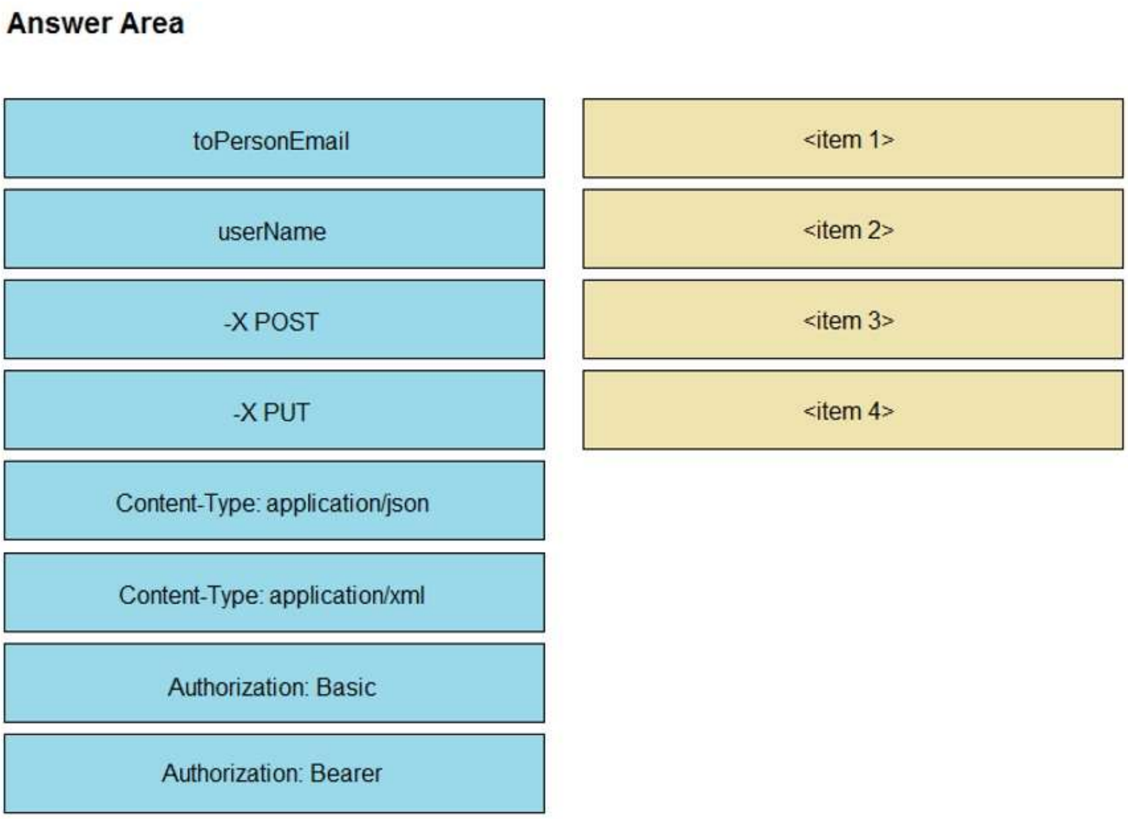
`toPersonEmail`
string
The email address of the recipient when sending a private 1:1 message.

`text`
string
The message, in plain text. If `markdown` is specified this parameter may be *optionally* used to provide alternate text for UI clients that do not support rich text. The maximum message length is 7439 bytes.

`markdown`
string
The message, in Markdown format. The maximum message length is 7439 bytes.



```
#!/bin/bash
curl <item 1> https://api.ciscopark.com/v1/messages \
     -H ‘<item 2>’ \
     -H ‘<item 3> NMU4NjQ0YWUtNjy_P..._1eb6574-ad72cae0e10f’\
     -d ‘{ “<item 4>”: “cisco@usa.net”, “text”: “Intruder Alert!” }’
```



Answer Area

toPersonEmail

userName

-X POST

-X PUT

Content-Type: application/json

Content-Type: application/xml

Authorization: Basic

Authorization: Bearer

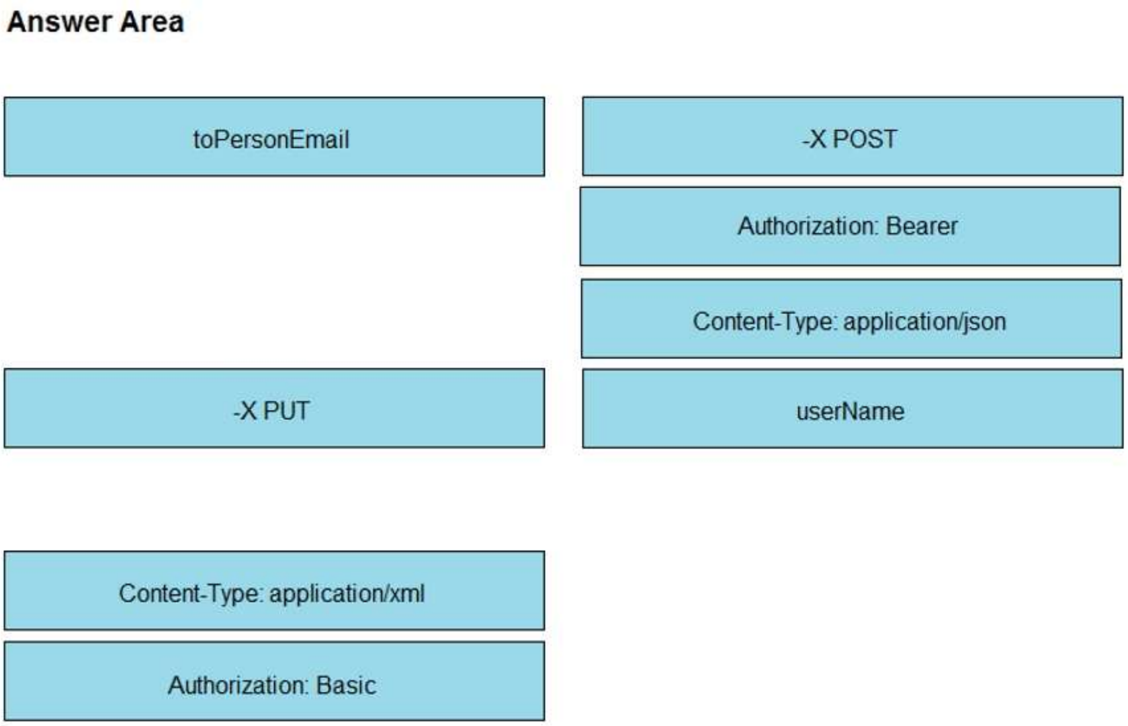
<item 1>

<item 2>

<item 3>

<item 4>

Answer:



Answer Area

toPersonEmail

-X PUT

Content-Type: application/xml

Authorization: Basic

-X POST

Authorization: Bearer

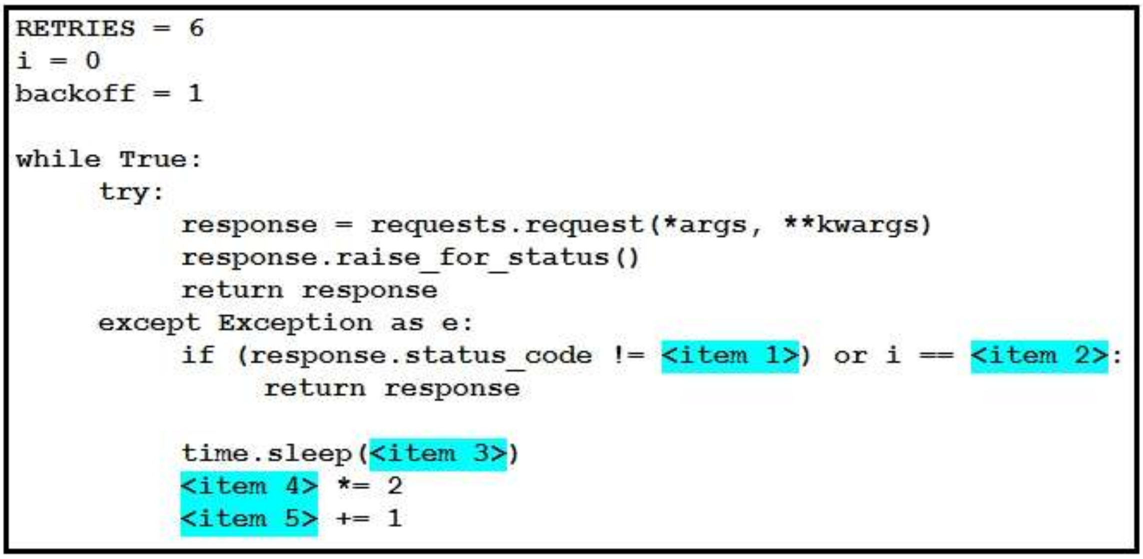
Content-Type: application/json

userName

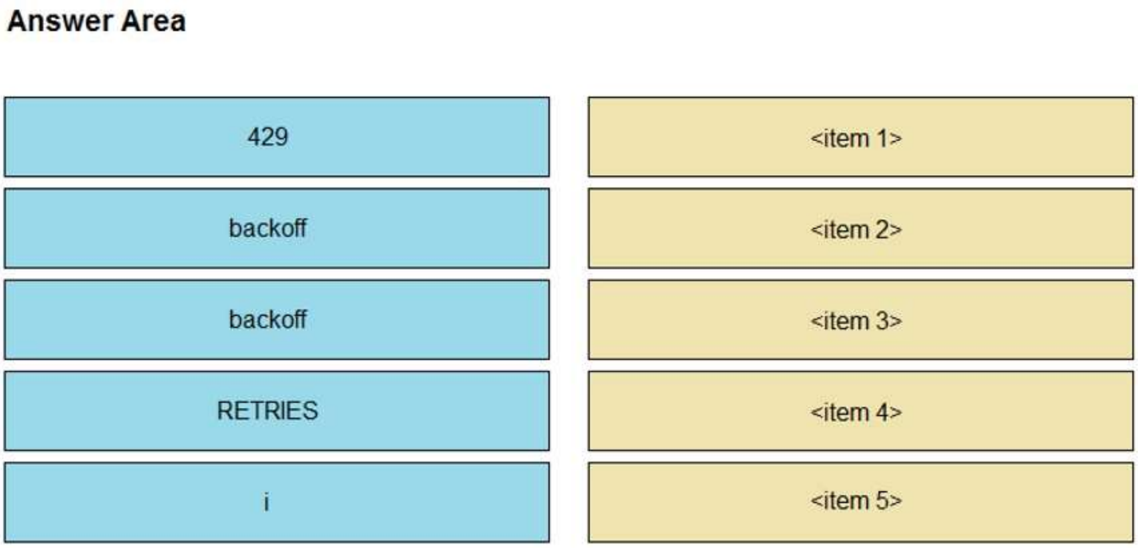
QUESTION 30

Drag and Drop Question

Refer to the exhibit. The self-service Webex Teams bot is failing when many users attempt to interact with it at the same time. Drag and drop the code snippets from the left onto the correct item numbers on the right that match the missing sections in the exhibit to complete this code to handle this high-load situation.



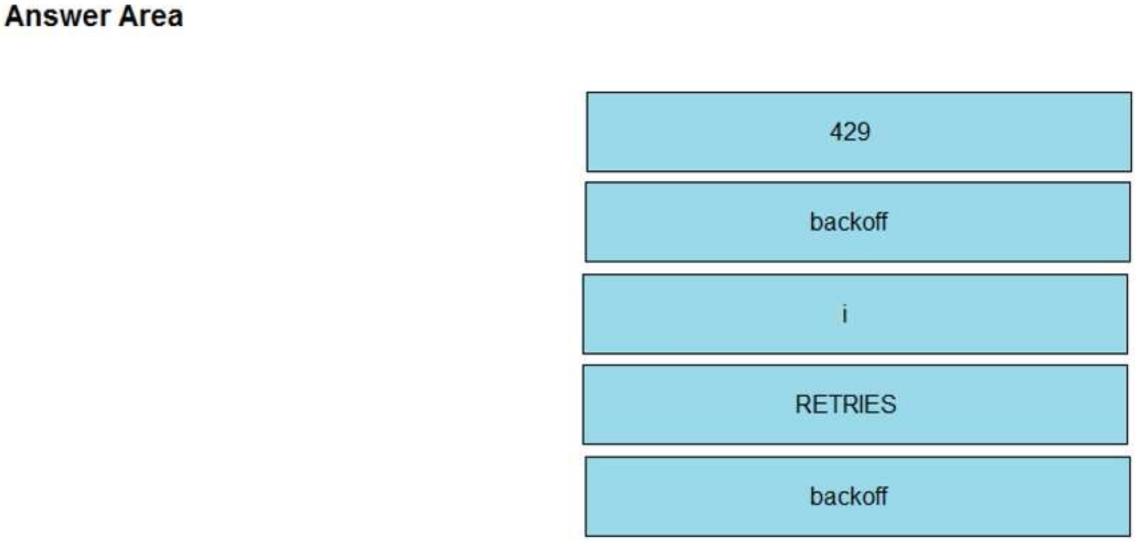
```
RETRIES = 6
i = 0
backoff = 1

while True:
     try:
          response = requests.request(*args, **kwargs)
          response.raise_for_status()
          return response
     except Exception as e:
          if (response.status_code != <item 1>) or i == <item 2>:
               return response

          time.sleep(<item 3>)
          <item 4> *= 2
          <item 5> += 1
```



Answer Area

429	<item 1>
backoff	<item 2>
backoff	<item 3>
RETRIES	<item 4>
i	<item 5>

Answer:



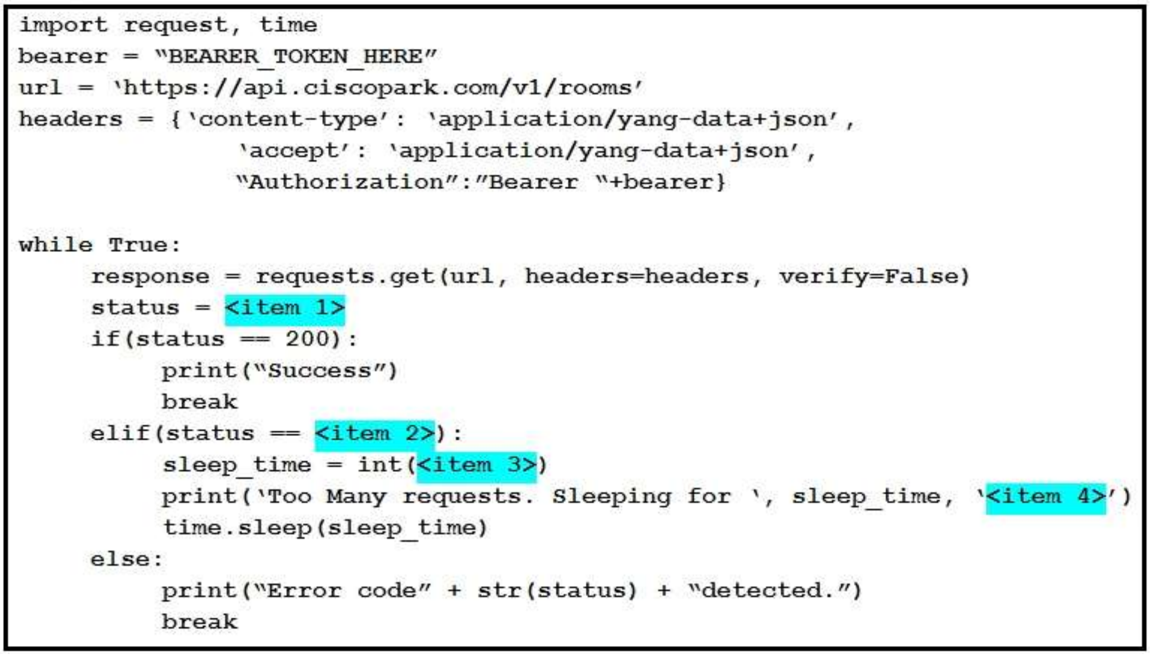
Answer Area

429
backoff
i
RETRIES
backoff

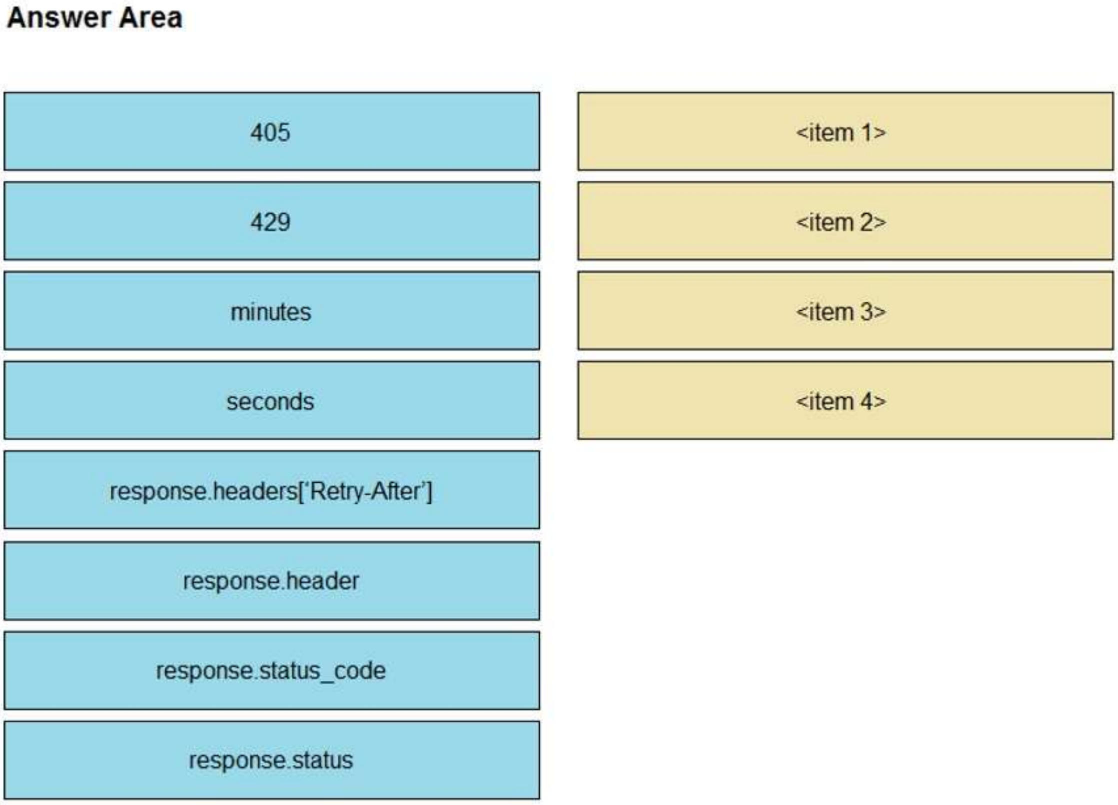
QUESTION 31

Drag and Drop Question

Refer to the exhibit. A developer is creating a Python script to use the Webex Teams REST API to list joined spaces, retry after the server-specified amount of time if a "Too many requests" response is received, and print any other error that is received. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the script. Not all code snippets are used.



```
import request, time
bearer = "BEARER_TOKEN_HERE"
url = 'https://api.ciscopark.com/v1/rooms'
headers = {'content-type': 'application/yang-data+json',
              'accept': 'application/yang-data+json',
              "Authorization":"Bearer "+bearer}

while True:
     response = requests.get(url, headers=headers, verify=False)
     status = <item 1>
     if(status == 200):
          print("Success")
          break
     elif(status == <item 2>):
          sleep_time = int(<item 3>)
          print('Too Many requests. Sleeping for ', sleep_time, '<item 4>')
          time.sleep(sleep_time)
     else:
          print("Error code" + str(status) + "detected.")
          break
```



Answer Area

405

429

minutes

seconds

response.headers['Retry-After']

response.header

response.status_code

response.status

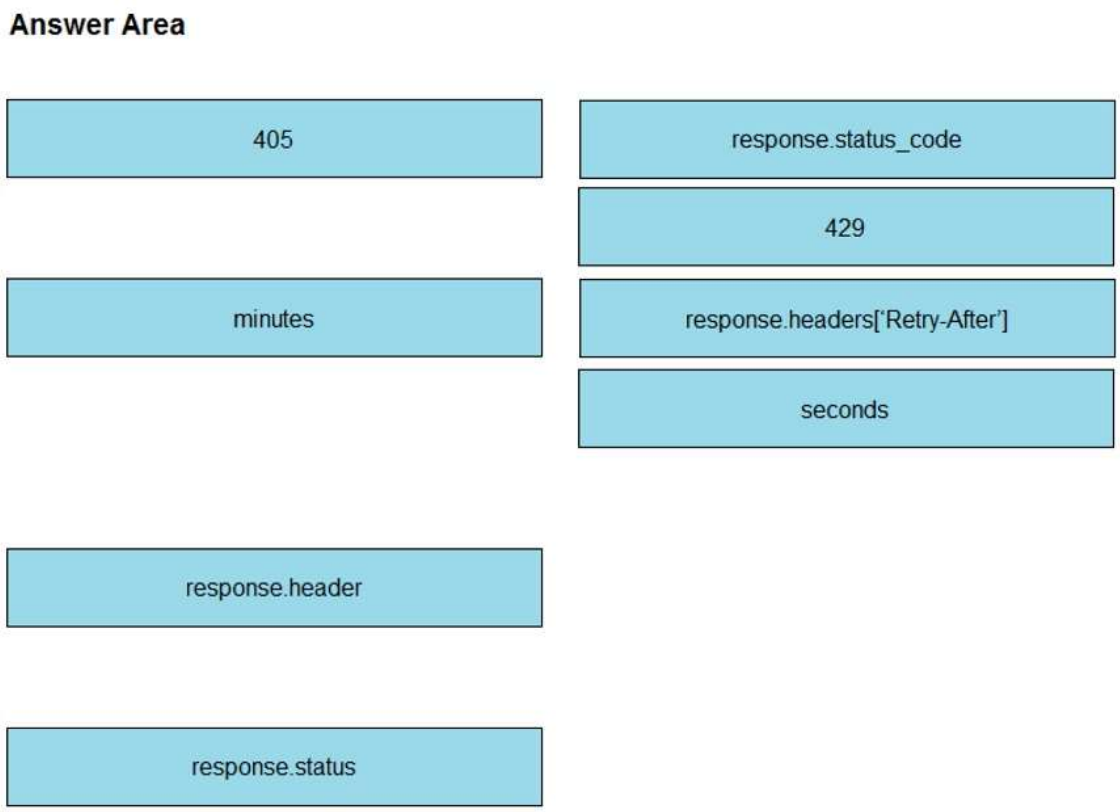
<item 1>

<item 2>

<item 3>

<item 4>

Answer:



Answer Area

405

response.status_code

429

minutes

response.headers['Retry-After']

seconds

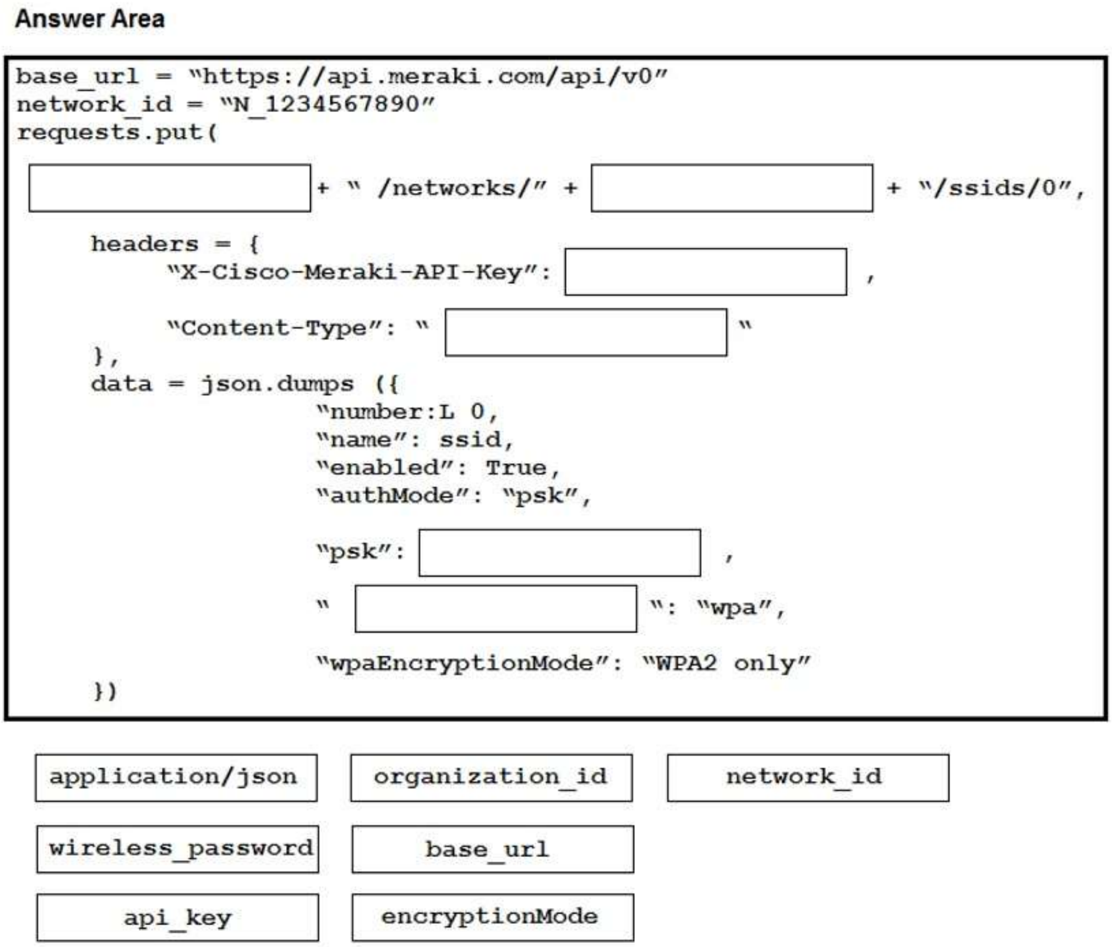
response.header

response.status

QUESTION 32
FILL BLANK
Fill in the blanks to complete the Python script to enable the SSID with a name of "371767916" in the network resource "11111111" using the Meraki Dashboard API.
Answer: See explanation below
Explanation:

1. 371767916
2. "{{HTTP_METHOD}}"
3. payload

QUESTION 33
Drag and Drop Question
Drag and drop the code onto the snippet to update a SSID in Meraki using the Dashboard API. Not all options are used.

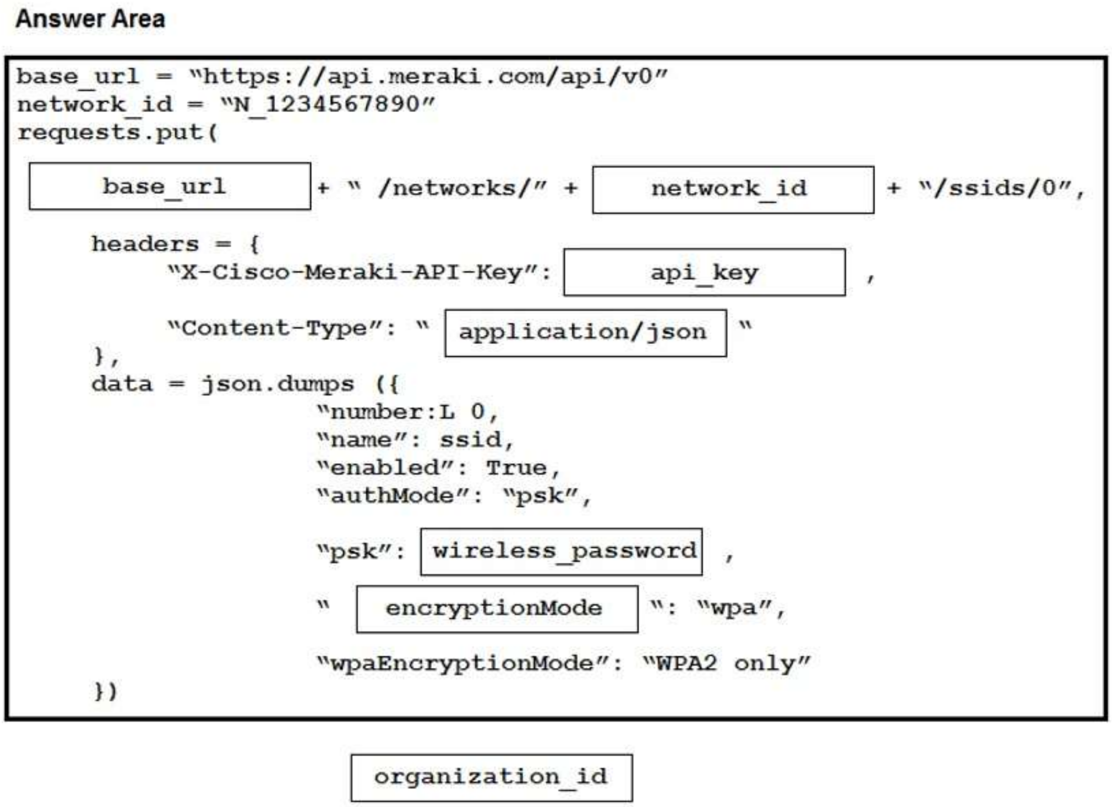


Answer Area

```
base_url = "https://api.meraki.com/api/v0"
network_id = "N_1234567890"
requests.put(

 __________ + " /networks/" + __________ + "/ssids/0",

     headers = {
          "X-Cisco-Meraki-API-Key": __________ ,

          "Content-Type": " __________ "
     },
     data = json.dumps ({
                         "number:L 0,
                         "name": ssid,
                         "enabled": True,
                         "authMode": "psk",

                         "psk": __________ ,

                         " __________ ": "wpa",

                         "wpaEncryptionMode": "WPA2 only"
     })
```

application/json	organization_id	network_id
wireless_password	base_url	
api_key	encryptionMode	

Answer:



Answer Area

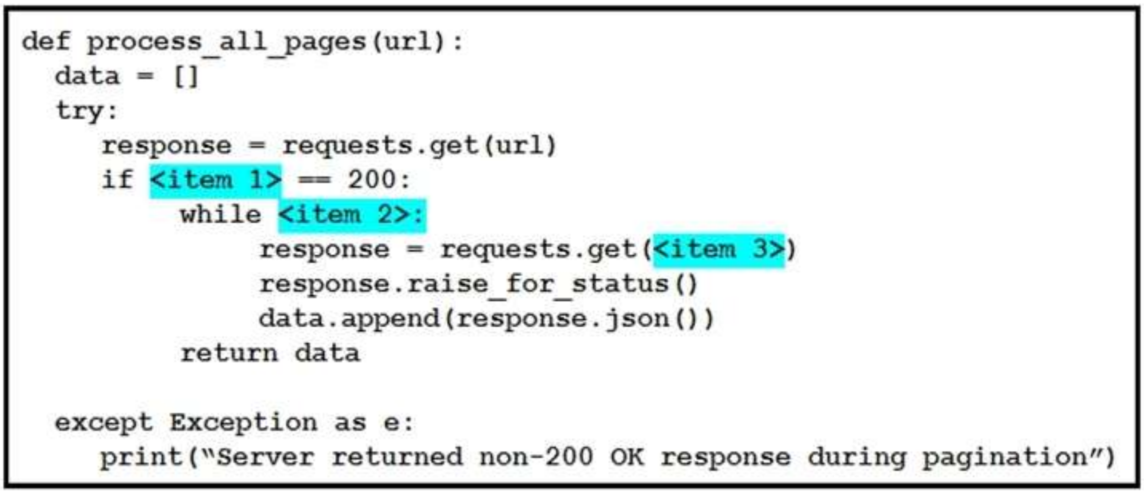
```
base_url = “https://api.meraki.com/api/v0”
network_id = “N_1234567890”
requests.put(

 base_url  + “ /networks/” + network_id  + “/ssids/0”,

    headers = {
        “X-Cisco-Meraki-API-Key”: api_key ,

        “Content-Type”: “ application/json ”
    },
    data = json.dumps ({
                    “number:L 0,
                    “name”: ssid,
                    “enabled”: True,
                    “authMode”: “psk”,

                    “psk”: wireless_password ,

                    “ encryptionMode ”: “wpa”,

                    “wpaEncryptionMode”: “WPA2 only”
    })
```

```
organization_id
```

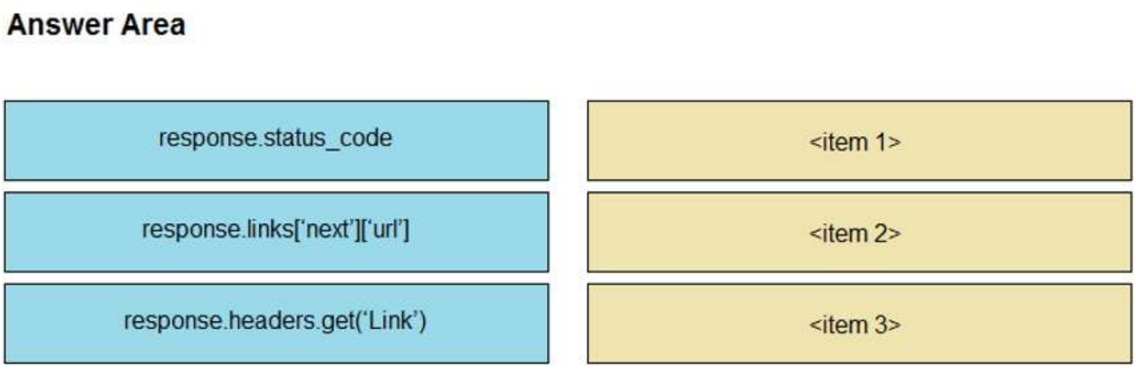
QUESTION 34

Drag and Drop Question

Refer to the exhibit. Drag and drop the parts of the Python code from the left onto the item numbers on the right that match the missing sections in the exhibit that consumes REST API pagination.



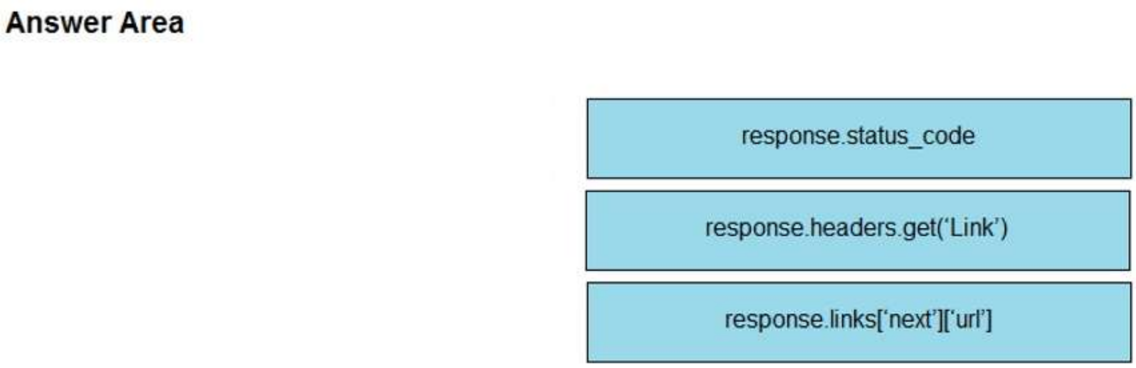
```
def process_all_pages(url):
  data = []
  try:
     response = requests.get(url)
     if <item 1> == 200:
          while <item 2>:
               response = requests.get(<item 3>)
               response.raise_for_status()
               data.append(response.json())
          return data

  except Exception as e:
     print(“Server returned non-200 OK response during pagination”)
```



Answer Area

response.status_code	<item 1>
response.links['next']['url']	<item 2>
response.headers.get('Link')	<item 3>

Answer:



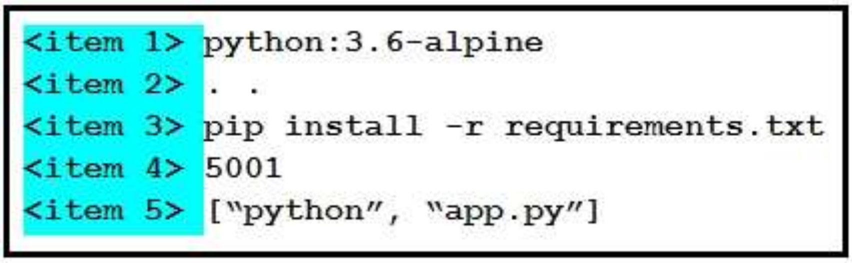
Answer Area

	response.status_code
	response.headers.get('Link')
	response.links['next']['url']

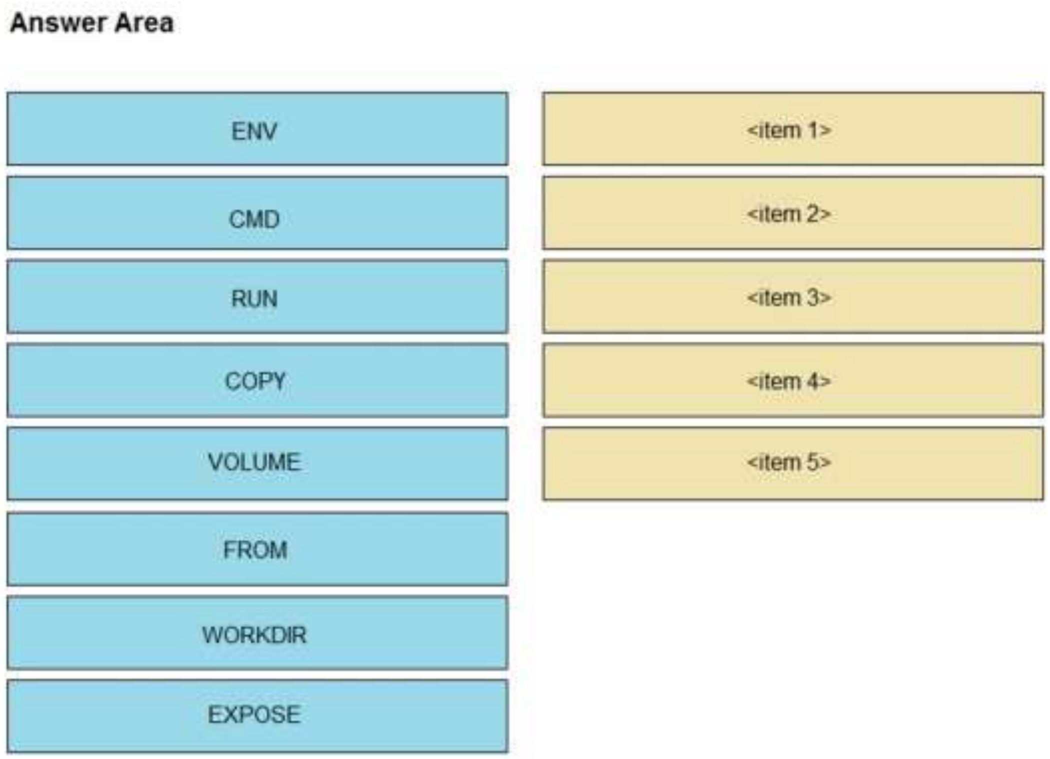
QUESTION 35

Drag and Drop Question

Refer to the exhibit. Drag and drop the correct parts of the Dockerfile from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the Dockerfile to successfully build and deploy a container running a Python application. Not all parts of the Dockerfile are used.



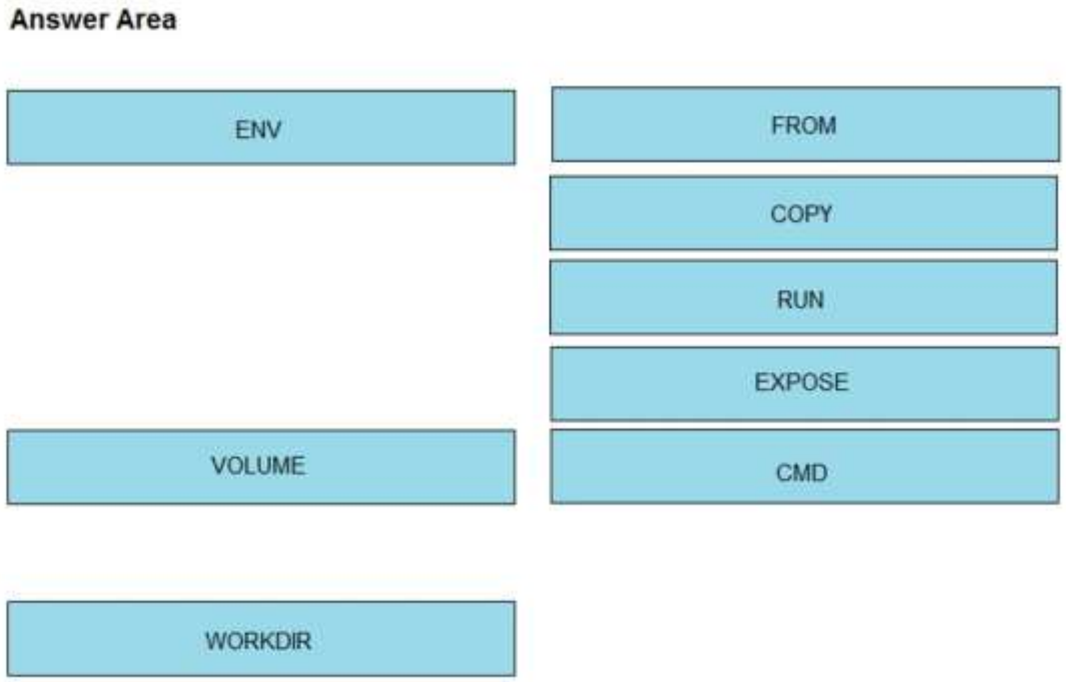
```
<item 1> python:3.6-alpine
<item 2> . .
<item 3> pip install -r requirements.txt
<item 4> 5001
<item 5> ["python", "app.py"]
```



Answer Area

ENV	<item 1>
CMD	<item 2>
RUN	<item 3>
COPY	<item 4>
VOLUME	<item 5>
FROM	
WORKDIR	
EXPOSE	

Answer:



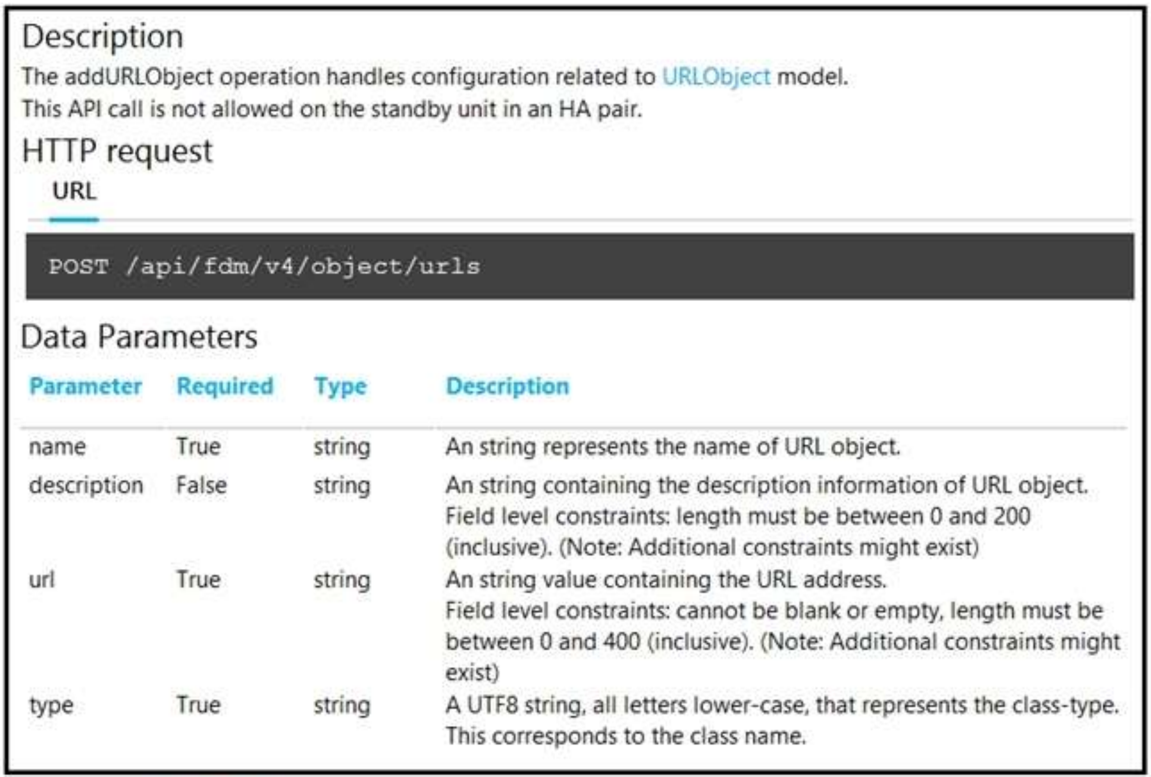
Answer Area

ENV	FROM
	COPY
	RUN
	EXPOSE
VOLUME	CMD
WORKDIR	

QUESTION 36

Drag and Drop Question

Refer to the exhibit. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the curl exhibit to complete the cURL request to FirePower Device Manager API to create objects. Not all code snippets are used.



Description

The addURLObject operation handles configuration related to URLObject model.
This API call is not allowed on the standby unit in an HA pair.

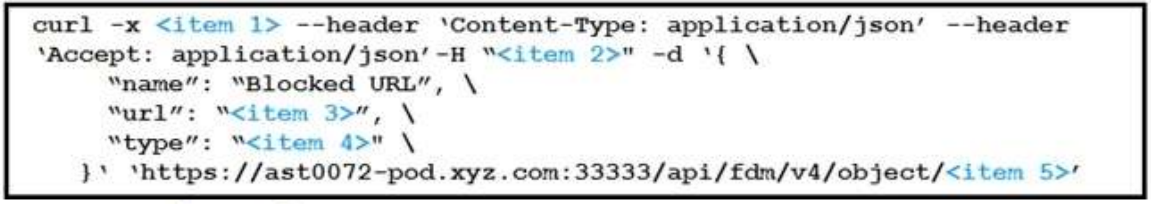
HTTP request

URL

```
POST /api/fdm/v4/object/urls
```

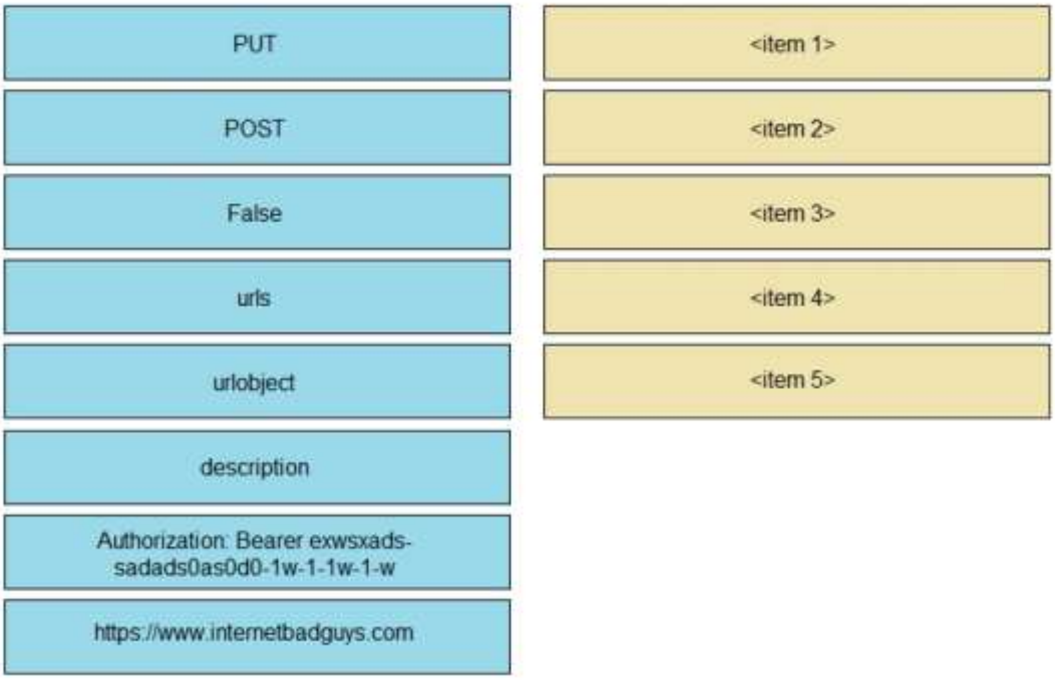
Data Parameters

Parameter	Required	Type	Description
name	True	string	An string represents the name of URL object.
description	False	string	An string containing the description information of URL object. Field level constraints: length must be between 0 and 200 (inclusive). (Note: Additional constraints might exist)
url	True	string	An string value containing the URL address. Field level constraints: cannot be blank or empty, length must be between 0 and 400 (inclusive). (Note: Additional constraints might exist)
type	True	string	A UTF8 string, all letters lower-case, that represents the class-type. This corresponds to the class name.



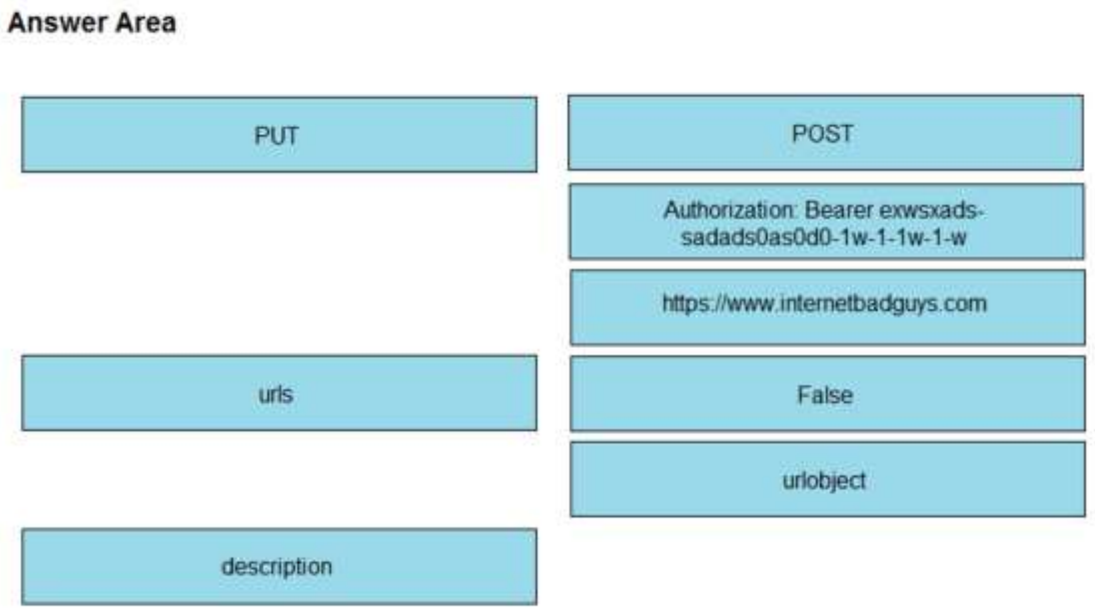
```
curl -x <item 1> --header 'Content-Type: application/json' --header
'Accept: application/json'-H "<item 2>" -d '{ \
     "name": "Blocked URL", \
     "url": "<item 3>", \
     "type": "<item 4>" \
   }' 'https://ast0072-pod.xyz.com:33333/api/fdm/v4/object/<item 5>'
```

Answer Area



PUT
POST
False
urls
urlobject
description
Authorization: Bearer exwsxads-sadads0as0d0-1w-1-1w-1-w
https://www.internetbadguys.com
<item 1>
<item 2>
<item 3>
<item 4>
<item 5>

Answer:



Answer Area

PUT

POST

Authorization: Bearer exwsxads-sadads0as0d0-1w-1-1w-1-w

https://www.internetbadguys.com

urls

False

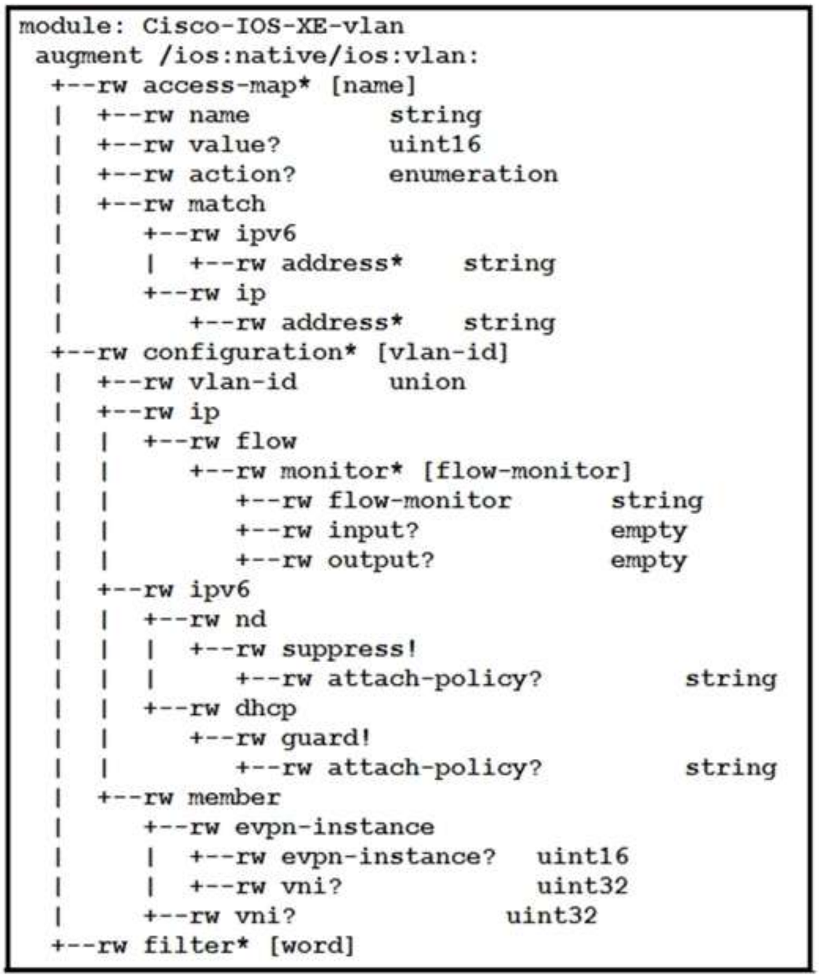
urlobject

description

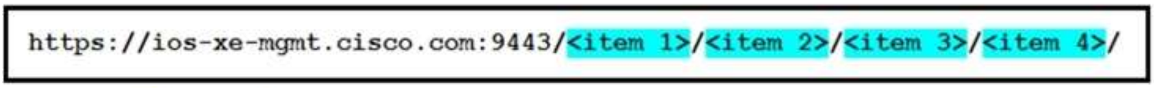
QUESTION 37

Drag and Drop Question

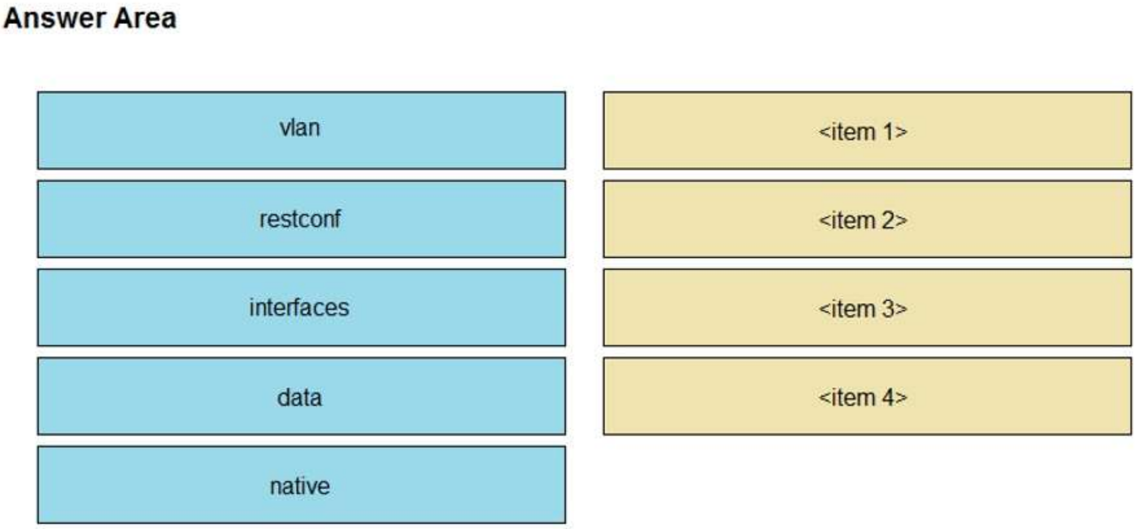
Refer to the exhibit. Drag and drop parts of the URL from the left onto the item numbers on the right that match the missing sections in the exhibit to create the appropriate RESTCONF URL to query the VLAN configuration given this YANG model. Not all URL parts are used.



```
module: Cisco-IOS-XE-vlan
 augment /ios:native/ios:vlan:
  +--rw access-map* [name]
  |  +--rw name          string
  |  +--rw value?        uint16
  |  +--rw action?       enumeration
  |  +--rw match
  |     +--rw ipv6
  |     |  +--rw address*     string
  |     +--rw ip
  |        +--rw address*     string
  +--rw configuration* [vlan-id]
  |  +--rw vlan-id       union
  |  +--rw ip
  |  |  +--rw flow
  |  |     +--rw monitor* [flow-monitor]
  |  |        +--rw flow-monitor       string
  |  |        +--rw input?             empty
  |  |        +--rw output?            empty
  |  +--rw ipv6
  |  |  +--rw nd
  |  |  |  +--rw suppress!
  |  |  |     +--rw attach-policy?          string
  |  |  +--rw dhcp
  |  |     +--rw guard!
  |  |        +--rw attach-policy?          string
  |  +--rw member
  |     +--rw evpn-instance
  |     |  +--rw evpn-instance?   uint16
  |     |  +--rw vni?             uint32
  |     +--rw vni?              uint32
  +--rw filter* [word]
```



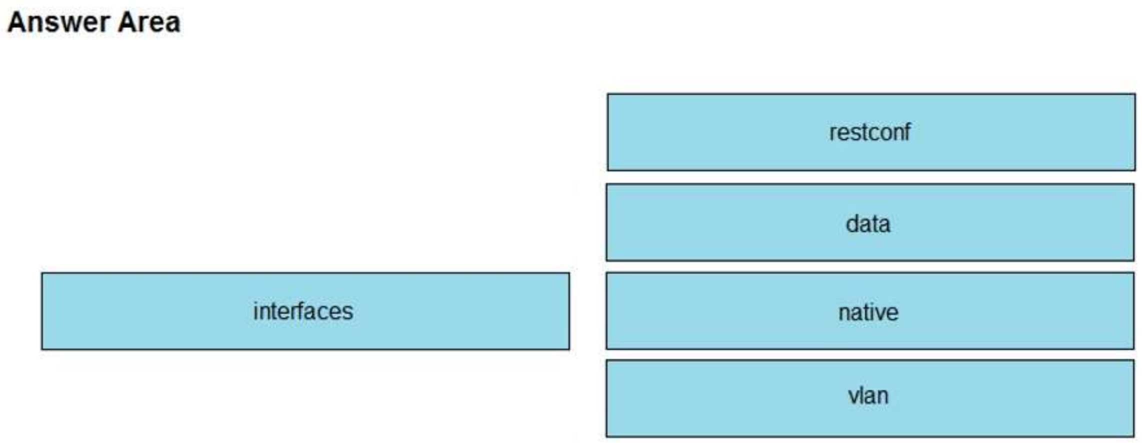
```
https://ios-xe-mgmt.cisco.com:9443/<item 1>/<item 2>/<item 3>/<item 4>/
```



Answer Area

vlan	<item 1>
restconf	<item 2>
interfaces	<item 3>
data	<item 4>
native	

Answer:



Answer Area

interfaces

restconf

data

native

vlan