



Developing Applications for Cisco Webex and Webex Devices (DEVWBX)



EXAMKILLER

Help Pass Your Exam At First Try

Cisco

Exam 300-920

Developing Applications for Cisco Webex and Webex Devices (DEVWBX)

Version: 3.0

[Total Questions: 60]

Question No : 1

A Webex Teams bot is deployed but soon it stops responding. Which two explanations are the cause of the issue? (Choose two.)

- A. A new webhook was created, which marks the old webhook as inactive.
- B. The web server that is set to receive webhooks is not configured to return a 200 message. And the webhook is disabled.
- C. The webhook secret is expired and must be refreshed.
- D. The refresh token is not being used.
- E. The bot owner regenerated the access token on developer.webex.com.

Answer: A,B

Reference: <https://developer.authorize.net/api/reference/features/webhooks.html>

Question No : 2

Refer to the exhibit.

```
const request = require('request');
request.post({url: 'https://api.webex.com/WBXService/XMLService',
  headers: { 'Content-Type': 'text/xml' },
  body : '<message><header><securityContext>
    <siteName>apidemoeu</siteName>
    <webExID>alice</webExID>
    <sessionTicket>AAABb6DpCJgAABUU2X0FMR09SSVRITV8=</sessionTicket>
  </securityContext></header>
  <body><bodyContent xsi:type= "java:com.webex.service.binding.meeting.CreateMeeting">
    <accessControl>
      <meetingPassword>Cisco1234</meetingPassword>
    </accessControl>
    <metaData><confName>Sample Meeting</confName>
    <meetingType>105</meetingType></metaData>
    <schedule><startDate>1/13/2020 16:00:00</startDate></schedule>
  </bodyContent></body></message>' },
  function (error, response, body) {
    if (!error && response.statusCode === 200) {
      console.log('Here!') } });
```

The Node.js script shown uses the Webex Meetings XML API to print "Here!" to the console. Which statement is a correct observation about the results of the script?

- A. The <meetingPassword> was not complex enough.
- B. The <sessionTicket> credential was expired.
- C. The WebexMeetings XML API service processed the request.
- D. The meeting was created successfully.

Answer: C**Explanation:**

The password, although not that good, has a capital letter and numbers. Therefore, it is okay. SessionTicket credential is not expired because the error function doesn't check that. We are not sure if the meeting was created successfully however, there is no wrong in the code, therefore, webexmeetings XML API service has processed the request.

Question No : 3

Which REST API request is used to list all the Webex Room Kit devices within a large organization so that a new custom In-Room Control can be deployed on all the devices?

A.

```
var request = require("request");
var options = { method: 'GET',
  url: 'https://api.ciscospark.com/v1/devices',
  qs: { product: 'Roomkit' },
  headers:
    { 'Content Type': 'application/json',
      Authorization: 'Bearer Yz6FgoWx7Pgb57C9z' } };

request(options, function(error, reponse, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

B.

```
var request = require("request");
var options = { method: 'GET',
  url: 'https://api.ciscospark.com/v1/devices',
  qs: { product: 'Roomkit' , placeID: 'Yzb60gRx3kBq5iB2w' },
  headers:
    { 'Content Type': 'application/json',
      Authorization: 'Bearer Yz6FgoWx7Pgb57C9z' } };

request(options, function(error, reponse, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

C.

```
var request = require("request");
var options = { method: 'GET',
  url: 'https://api.ciscospark.com/v1/devices/Yzb60gRx3kBq5iB2w',
  qs: { deviceName: 'Roomkit' },
  headers:
    { 'Content Type': 'application/json',
      Authorization: 'Bearer Yz6FgoWx7Pgb57C9z' } };

request(options, function(error, reponse, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

D.

```
var request = require("request");
var options = { method: 'GET',
  url: 'https://api.ciscospark.com/v1/devices',
  qs: { upgradeChannel: 'Roomkit' },
  headers:
    { 'Content Type': 'application/json',
      Authorization: 'Bearer Yz6FgoWx7Pgb57C9z' } };

request(options, function(error, reponse, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

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

Answer: A**Explanation:**

The qs: option is required to list all roomkit devices. Product: 'RoomKit' is the correct option because it will list all roomkit devices.

Question No : 4 DRAG DROP

Drag and drop the expressions to create a Webex Teams widget that uses Guest Issuer to enable customers to chat and meet with agents. Not all options are used.

```
<script>
$(document).ready(function () {
  $("#btn").click(function () {
    var widgetEL = document.getElementById('supportChart');
    $.ajax({
      url: '/api/getSession',
      type: 'POST',
      contentType: 'application/json',
      data: JSON.stringify({
        name: $("#name").val(),
        email: $("#email").val()
      })
    }).done(function (session) {
      ciscospark.widget(widgetEl).spaceWidget({
        guestToken: ,
        destinationId: ,
        destinationType: ,
        activities: {"files": false, "meet": true, "message": true, "people": false},
        initialActivity: ,
        secondaryActivitiesFullWidth: false
      });
    })
  })
})
</script>

app.post('/api/getSession', function(req, res) {
  res.status(200).send({
    agent: 'johndoe@example.com',
    token: jwt.sign({
      "sub": "${req.body.email}",
      "name": "${req.body.name}",
      "iss": "${issuer}"
    }, Buffer.from("${secret}", 'base64'), {})
  });
});
})
```

“userId”
 “message”
 session.token
 Direct
 session.agent
 guest

Answer:

```

<script>
$(document).ready(function () {
  $("#btn").click(function () {
    var widgetEL = document.getElementById('supportChart');
    $.ajax({
      url: '/api/getSession',
      type: 'POST',
      contentType: 'application/json',
      data: JSON.stringify({
        name: $("#name").val(),
        email: $("#email").val()
      })
    }).done(function (session) {
      ciscospark.widget(widgetEL).spaceWidget({
        guestToken: guest,
        destinationId: "userId",
        destinationType: session.token,
        activities: {"files": false, "meet": true, "message": true, "people": false},
        initialActivity: session.agent,
        secondaryActivitiesFullWidth: false
      });
    })
  })
})
</script>

app.post('/api/getSession', function(req, res) {
  res.status(200).send({
    agent: 'johndoe@example.com',
    token: jwt.sign({
      "sub": "${req.body.email}",
      "name": "${req.body.name}",
      "iss": "${issuer}"
    }, Buffer.from("${secret}", 'base64'), {})
  });
});

```

"userId"

"message"

session.token

Direct

session.agent

guest

Question No : 5 DRAG DROP

Drag and drop the code segments from the right of the screen into the targets on the left to create a script that retrieves a list of rooms using the Webex JavaScript API and outputs the title and ID to the console for one that matches the title "My Webex discussion". Not all code segments are used.

```
var Webex = require("webex")
var webex = Webex.init({
  credentials: {access_token: 'jhgdhdiuytiut-hgjhgf'}
});
```

```
    console.log(room.title.room.id)
  })
  catch(function(reason) {
    console.error(reason);
    process.exit(1);
  })
})
```

webex.rooms

```
.list({
  max:500
})
```

```
var room = rooms.items.filter(function(room){
  return title.room == "My Webex Discussion"
});
```

```
.resolve(function(rooms){
```

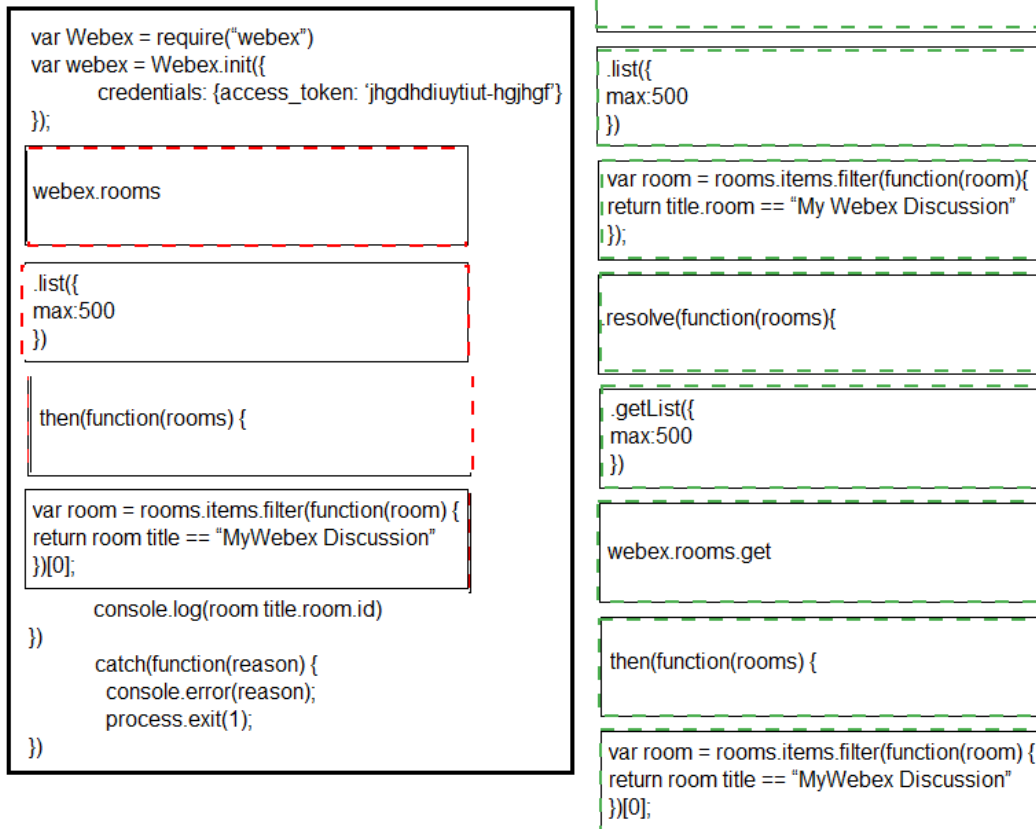
```
.getList({
  max:500
})
```

webex.rooms.get

```
then(function(rooms) {
```

```
var room = rooms.items.filter(function(room) {
  return room.title == "MyWebex Discussion"
})[0];
```

Answer:

**Question No : 6**

What happens if a meeting is in progress when a DelMeeting request is sent in the Webex Meetings XML API?

- A. The meeting host is notified and prompted to allow the meeting to be deleted.
- B. The DelMeeting request drops all call-in users and deletes the meeting.
- C. The DelMeeting request waits until the meeting is completed and then deletes the meeting.
- D. The DelMeeting request results in an error.

Answer: A

Question No : 7

Refer to the exhibit.

```
xcommand UserInterface Extensions List
OK
*r ExtensionsListResult (status=OK):
*r ExtensionsListResult Extensions Version: "1.5"
*r ExtensionsListResult Extensions Panel 1 Icon: Lightbulb
*r ExtensionsListResult Extensions Panel 1 Type: Statusbar
*r ExtensionsListResult Extensions Panel 1 Name: Panel
*r ExtensionsListResult Extensions Panel 1 PanelId: "panel_1"
*r ExtensionsListResult Extensions Panel 1 Order: 1
*r ExtensionsListResult Extensions Panel 1 Page 1 Name: "Page"
*r ExtensionsListResult Extensions Panel 1 Page 1 Row 1 Name: "Row"
*r ExtensionsListResult Extensions Panel 1 Page 1 Row 1 Widget 1 WidgetId: "widget_1"
*r ExtensionsListResult Extensions Panel 1 Page 1 Row 1 Widget 1 Type: Slider
*r ExtensionsListResult Extensions Panel 1 Page 1 Row 1 Widget 1 Options: "size=3"
** end
--
```

With this in-room configuration, which command successfully sets the slider to 50?

- A. xcommand UserInterface Extensions Widget SetSlider Value: 50
- B. xcommand UserInterface Extensions Widget Slider: 50 Widget: "widget_1"
- C. xconfiguration UserInterface Extensions widget_1 50
- D. xcommand UserInterface Extensions Widget SetValue WidgetId: "widget_1" Value: 50

Answer: A

Reference: <https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/ce98/sx-mx-dx-room-kit-boards-customization-guide-ce98.pdf>

Question No : 8 DRAG DROP

Drag and drop the items to complete the jsxapi script. Not all options are used.

```
const xapi = require ( '  ');  
function printHttp(url) {  
xapi.  ('HTTPClient Get'. {Url: url })  
    .then((data) => {  
        console.log({  
            data:JSON.parse.body(data),  
        });  
    })  
    .  (() => {  
        console.error({  
            error: 'HTTP Get failed'.  
        });  
    });  
}  
 ('http://example.com');
```

Answer:

```
const xapi = require ( '  ');  
function printHttp(url) {  
xapi.  ('HTTPClient Get'. {Url: url })  
  .then((data) => {  
    console.log({  
      data:JSON.parse.body(data),  
    });  
  })  
  .  (() => {  
    console.error({  
      error: 'HTTP Get failed'.  
    });  
  });  
}  
 ('http://example.com');
```

Question No : 9 DRAG DROP

Drag and drop the code to complete the JavaScript code snippet to create a meeting using the Webex Meetings XML API. Options may be used more than once.

```

<script>
const init = {
  method: 'POST'
  body: <message xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
    <header><securityContext>
      <siteName>apidemoeu</siteName>
      <webExID>alice</webExID>
      <[redacted]>AAABb6FFuSBR19BRERJTkdfU0hBMjU2X0FMR09SSVRITV8=
    </[redacted]>
    </securityContext></header>
    <body>
      <bodyContent xsi:type="*ava:com.webex.service.binding.meeting.CreateMeeting">
        <accessControl><[redacted]>Cisco1234</[redacted]>
      </accessControl>
      <metaData><confName>Sample Meeting</confName>
      <meetingType>105</meetingType></metaData>

      <schedule><startDate>[redacted]</startDate></schedule>
    </bodyContent>
  </body></message>' }
fetch('https://api.webex.com/WBXService/XMLService', init)
  .then(response => response.text())
  .then(str => (new window.DOMParser()).parseFromString(str, '[redacted]'))
  .then(data => document.write(data.getElementsByTagName('*', '[redacted]')
[0].textContent))
</script>

```

result	application/json
text/xml	1/13/2020 16:00:00
meetingPassword	sessionTicket
Jan 24, 2019 4:00 PM	status

Answer:

```

<script>
const init = {
  method: 'POST'
  body: <message xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
    <header><securityContext>
      <siteName>apidemoeu</siteName>
      <webExID>alice</webExID>
      <[redacted sessionTicket]>AAABb6FFuSBR19BRERJTkdfU0hBMjU2X0FMR09SSVRITV8=
    </[redacted status]>
    </securityContext></header>
    <body>
      <bodyContent xsi:type="*ava:com.webex.service.binding.meeting.CreateMeeting">
        <accessControl><[redacted meetingPassword]>Cisco1234</[redacted meetingPassword]>
      </accessControl>
      <metaData><confName>Sample Meeting</confName>
      <meetingType>105</meetingType></metaData>

      <schedule><startDate>[redacted Jan 24, 2019 4:00 PM]</startDate></schedule>
    </bodyContent>
  </body></message>' }
fetch('https://api.webex.com/WBXService/XMLService', init)
  .then(response => response.text())
  .then(str => (new window.DOMParser()).parseFromString(str, '[redacted text/xml]'))
  .then(data => document.write(data.getElementsByTagName('*', '[redacted result]')
[0].textContent))
</script>

```

result	application/json
text/xml	1/13/2020 16:00:00
meetingPassword	sessionTicket
Jan 24, 2019 4:00 PM	status

Question No : 10 DRAG DROP

Refer to the exhibit.

```
{messages
  { "$":
    { 'xmlns:serv': 'http://www.webex.com/schemas/2002/06/service',
      'xmlns:com': 'http://www.webex.com/schemas/2002/06/common',
      'xmlns:ep': 'http://www.webex.com/schemas/2002/06/service/ep',
      'xmlns:meet': 'http://www.webex.com/schemas/2002/06/service/meeting' },
    header: { response: { result: 'SUCCESS', gsbStatus: 'PRIMARY' } },
    body:
      { bodyContent:
        { '$':
          { 'xsi:type': 'eplstRecordingResponse',
            'xmlns:xsi': 'http://www.w3.org/2001/XMLSchema-instance',
            matchingRecords: { total: '2', returned: '2', startFrom: '1' },
            recording:
              [ { recordingID: '90812441',
                  hostWebExID: 'jlev@cdpneighbors.com',
                  name: "Jeff Levesailor's Personal Room-20190711 1154-1",
                  createTime: '07/11/2019 08:06:27',
                  timeZoneID: '11',
                  size: '0.017041206',
                  streamURL: 'https://cdp.webex.com/wr/ldr.php?RCID=e7eba83fadfec6f5abd98637407499f7',
                  fileURL: 'https://cdp.webex.com/wr/lr.php?RCID=dc37fb81c5ba41e42db43247ac95b7e8',
                  recordingType: '5',
                  duration: '9',
                  format: 'MP4',
                  serviceType: 'MeetingCenter',
                  password: '4Mt3J3r4',
                  passwordReq: 'true',
                  confID: '133167544940915779',
                  shareToMe: 'false' },
                { recordingID: '90812244',
                  hostWebExID: 'jlev@cdpneighbors.com',
                  name: "Jeff Levensailor's Personal Room-20190711 1156-2",
                  createTime: '07/11/2019 08:06:26',
                  timeZoneID: '11',
                  size: '0.01679802',
                  streamURL: 'https://cdp.webex.com/wr/ldr.php?RCID=caf77cc9a8f564daeb06c6747c6eda01',
                  fileURL: 'https://cdp.webex.com/wr/lr.php?RCID=9a016ae489fe94af45b474c010047a81',
                  recordingType: '5',
                  duration: '6',
                  format: 'MP4',
                  serviceType: 'MeetingCenter',
                  password: 'vHJpXCr4',
                  passwordReq: 'true',
                  confID: '133167544940915779',
                  shareToMe: 'false' } ] } } } }
```

A training coordinator must post links to Webex recordings on a company SharePoint site. This is usually a manual process, but a DevOps engineer wants to automate it using Webex XML APIs. After a successful LstRecording call wrapped in xml2js, the 'console.dir(result)' output is shown in the exhibit. Using 'dot notation', drag and drop the code below onto the code snippet to output the streamURL for each recording.

```
'recordings = result ({ [ ] } { [ ] } { [ ] } { [ ] } )  
for (i=0, i<recordings.length, i++){Console.log(recordings[i] ({ [ ] })))}
```

body

recording

bodyContent

message

streamURL

Answer:

```
'recordings = result ({ body } { bodyContent } { streamURL } { message } )  
for (i=0, i<recordings.length, i++){Console.log(recordings[i] ({ recording })))}
```

body

recording

bodyContent

message

streamURL

Question No : 11

Refer to the exhibit.

```
{  
  "message": "The user has sent too many requests in a given amount of time. Please refer to the  
Retry-After response header to wait before making a new request.",  
  "errors": [  
    {  
      "description": "The user has sent too many requests in a given amount of time. Please  
refer to the Retry-After response header to wait before making a new request."  
    }  
  ],  
  "trackingId": "ROUTER_5D224C5A-C32B-01BB-007E-2D30B4D4007E"  
}
```

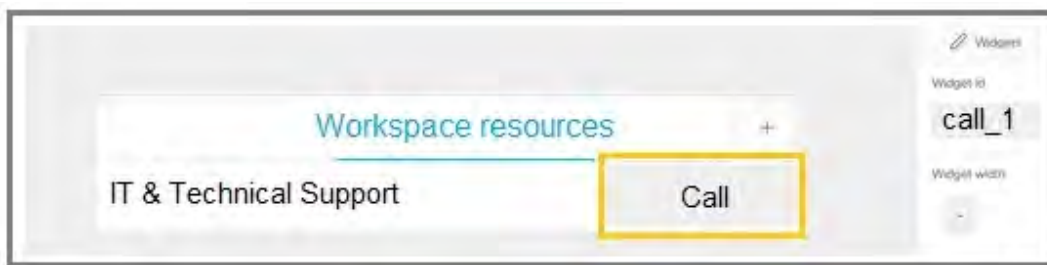
What is the Webex Teams REST API HTTP response status code, based on this code snippet?

- A. 401
- B. 403
- C. 429
- D. 501

Answer: CReference: <https://developer.webex.com/docs/api/v1/messages/get-message-details>

Question No : 12 DRAG DROP

Refer to the exhibit.



A Webex device In-Room Control editor screenshot and associated Macro code is shown. Drag and drop the code snippets to complete the JavaScript Macro that launches a call when the Call button on the custom control panel is touched. Not all options are used.

```
const xapi = require('xapi');

xapi.event.on(' ', (event)=> {
  if (event.WidgetId !== ' ') return;
  if (event.Type !== 'clicked') return;
  xapi.
  (' ', (Number: '1000'));
});
```

UserInterface InRoomControl Button Event

call_1

execute

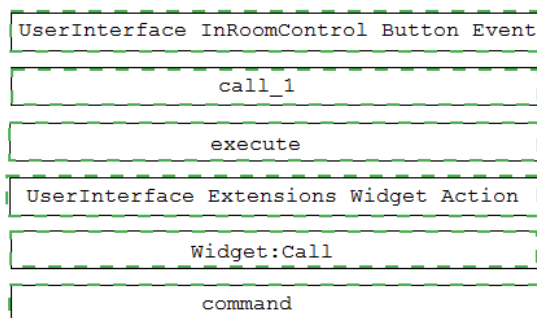
UserInterface Extensions Widget Action

Widget:Call

command

Answer:


```
const xapi = require('xapi');  
xapi.event.on('UI InRoomControl Button Event', (event)=> {  
  if (event.WidgetId !== 'execute') return;  
  if (event.Type !== 'clicked') return;  
  xapi.  
    (  
    'UI Extensions Widget Action', (Number: '1000'}));  
});
```

**Question No : 13**

With CE 9.8 and above, which two statements are correct when an application is sending and receiving data over a connection established with an xAPI interface? (Choose two.)

- A. All Serial, SSH, and WebSockets can be used to send and receive data.
- B. HttpClient can be used to send requests but not receive responses.
- C. HttpFeedback is the only option to receive data.
- D. The HttpClient command can be used to send requests and read responses over HTTP.
- E. WebSockets is the only option to send and receive data.

Answer: B,D

Reference: <https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/ce98/sx-mx-dx-room-kit-boards-customization-guide-ce98.pdf>

Question No : 14

Refer to the exhibit.

```
xapi.on('ready', () => {  
  console.log("connection successful");  
  
  // Start a call  
  
  * _____ *  
  
  .then((call) => {  
    console.log('Started call with status: ${call.status}, id: ${call.CallId}');  
  });  
});
```

Assume that there is already a connection to a device for this jsxapi.code snippet. Which snippet completes the code so that the connected device makes a call to the user jdoe@example.com?

- A. `xapi.command('Dial', {Endpoint: 'jdoe@example.com'})`
- B. `xapi.command('Call', { SIP: 'jdoe@example.com'})`
- C. `xapi.command('SIP', {Number: 'jdoe@example.com'})`
- D. `xapi.command('Dial', {Number: 'jdoe@example.com'})`

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

Answer: D

Reference: <https://github.com/CiscoDevNet/labs-xapi/pull/12/files/40c9fc38e115f4dec4709836501a165008caeaba>

Question No : 15

Refer to the exhibit.

```

1  const jsxapi = require('jsxapi');
2  const speaker = require('./speaker')
3  const xapi = jsxapi.connect('ssh://10.131.202.230', {
4      username: 'admin',
5      password: ''
6  });
7  let state = 0
8  function toggle(state) {
9      if (state === 0) {return 1}
10     else {return 0}
11 }
12 xapi.on('ready', () => {
13     xapi.event
14     .on('...', (x) => {
15         if (x.PanelId === 'panel_1') {
16             state = toggle(state)
17             if (state === 1) {
18                 speaker.control({"on": true});
19             } else {
20                 speaker.control({"on": false})
21             }
22         }
23     })
24 });

```

An 'Action Button' with identifier 'panel_1' is deployed to the Touch10 interface of a Room Series device. Which event must be inserted into line 14 to turn the music on/off in the conference room?

- A. UserInterface Extensions Event Pressed
- B. UserInterface Extensions Panel Clicked
- C. UserInterface Extensions Widget GetValue
- D. UserInterface Extensions Widget Action

Answer: A

Reference: <https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/ce81/sx-mx-in-room-control-guide-ce81.pdf>

Question No : 16 DRAG DROP

Drag and drop the definitions from the left onto the correct Webex Teams REST API HTTP response status codes on the right.

Too many requests have been sent in a given amount of time, and the request has been rate limited. A Retry-After header that specifies how many seconds you need to wait before a successful request can be made should be present.	401
The request to a resource used an unsupported HTTP request method.	403
Authentication credentials were missing or incorrect.	405
The request is understood but has been refused, or access is not allowed.	429

Answer:

Too many requests have been sent in a given amount of time, and the request has been rate limited. A Retry-After header that specifies how many seconds you need to wait before a successful request can be made should be present.	Authentication credentials were missing or incorrect.
The request to a resource used an unsupported HTTP request method.	The request is understood but has been refused, or access is not allowed.
Authentication credentials were missing or incorrect.	The request to a resource used an unsupported HTTP request method.
The request is understood but has been refused, or access is not allowed.	Too many requests have been sent in a given amount of time, and the request has been rate limited. A Retry-After header that specifies how many seconds you need to wait before a successful request can be made should be present.

Question No : 17

Refer to the exhibit.