# 300-920 $^{\text{Q\&As}}$

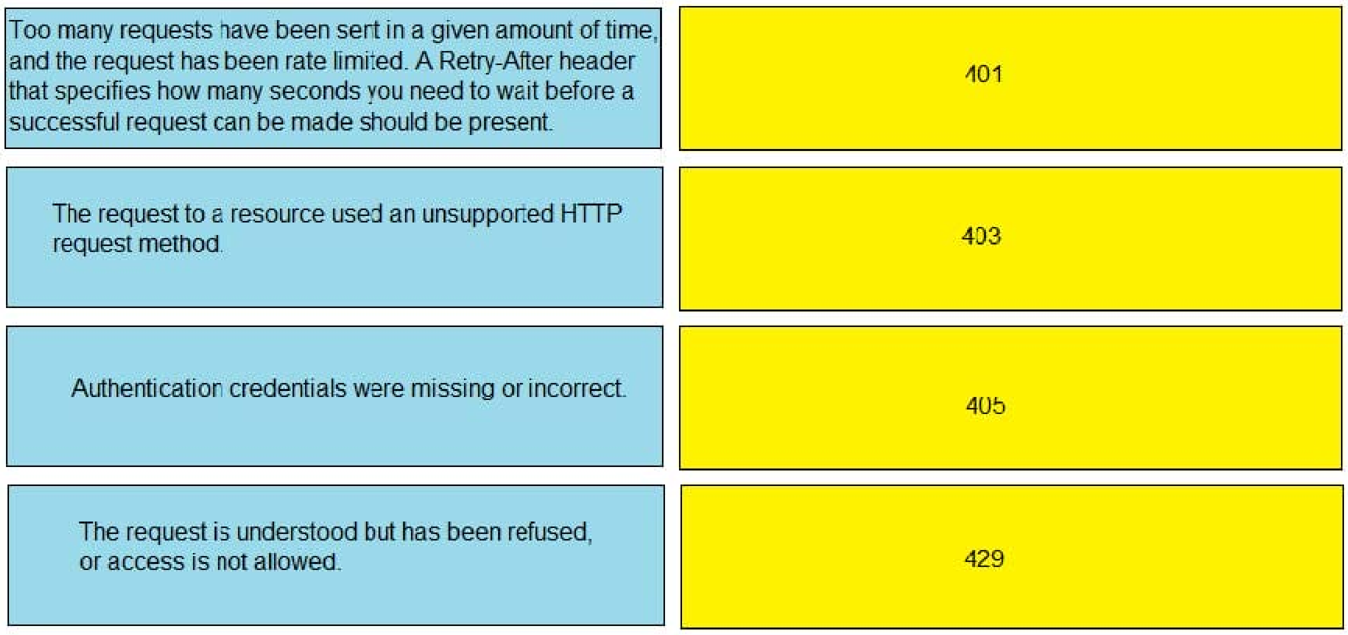
Developing Applications for Cisco Webex and Webex Devices (DEVWBX)

**QUESTION 1**

DRAG DROP

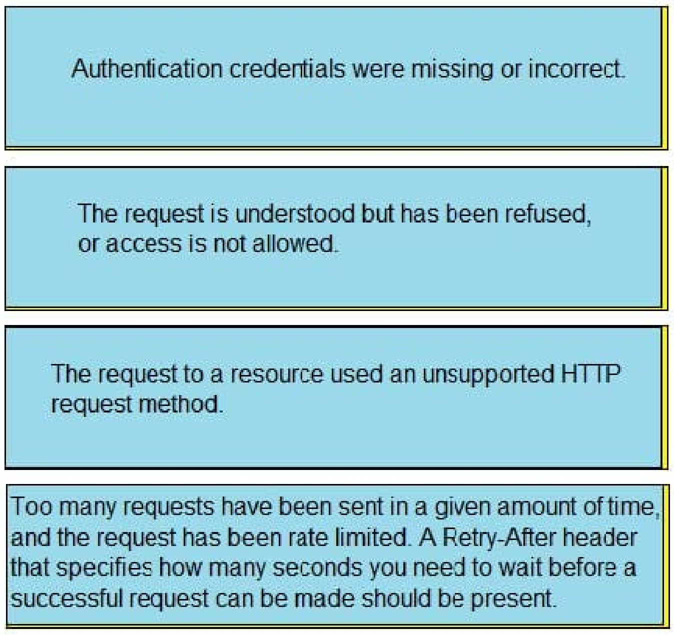
Drag and drop the definitions from the left onto the correct Webex Teams REST API HTTP response status codes on the right.

Select and Place:



| | |
|---|---|
| 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 |

Correct Answer:



Authentication credentials were missing or incorrect.

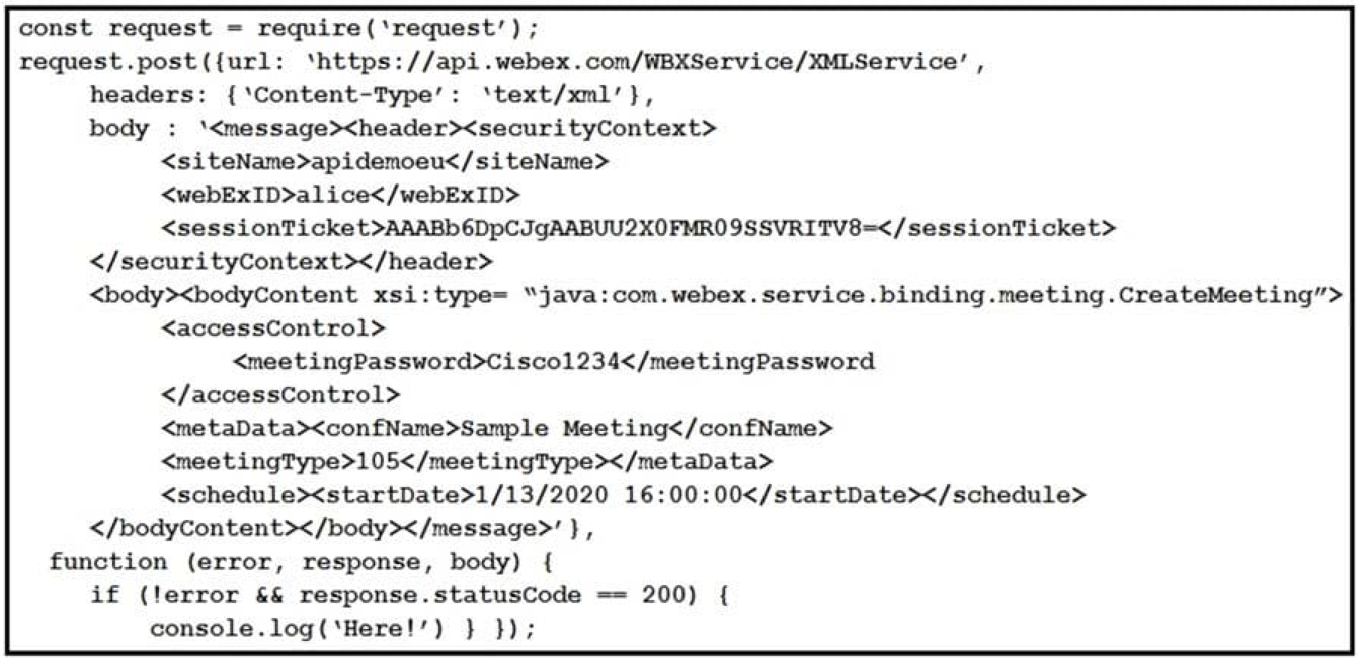
The request is understood but has been refused, or access is not allowed.

The request to a resource used an unsupported HTTP request method.

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.

Reference: https://developer.webex.com/docs/api/basics

**QUESTION 2**



```
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!’) } });
```

Refer to the exhibit. 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 was not complex enough.

B. The credential was expired.

C. The WebexMeetings XML API service processed the request.

D. The meeting was created successfully.

Correct Answer: C

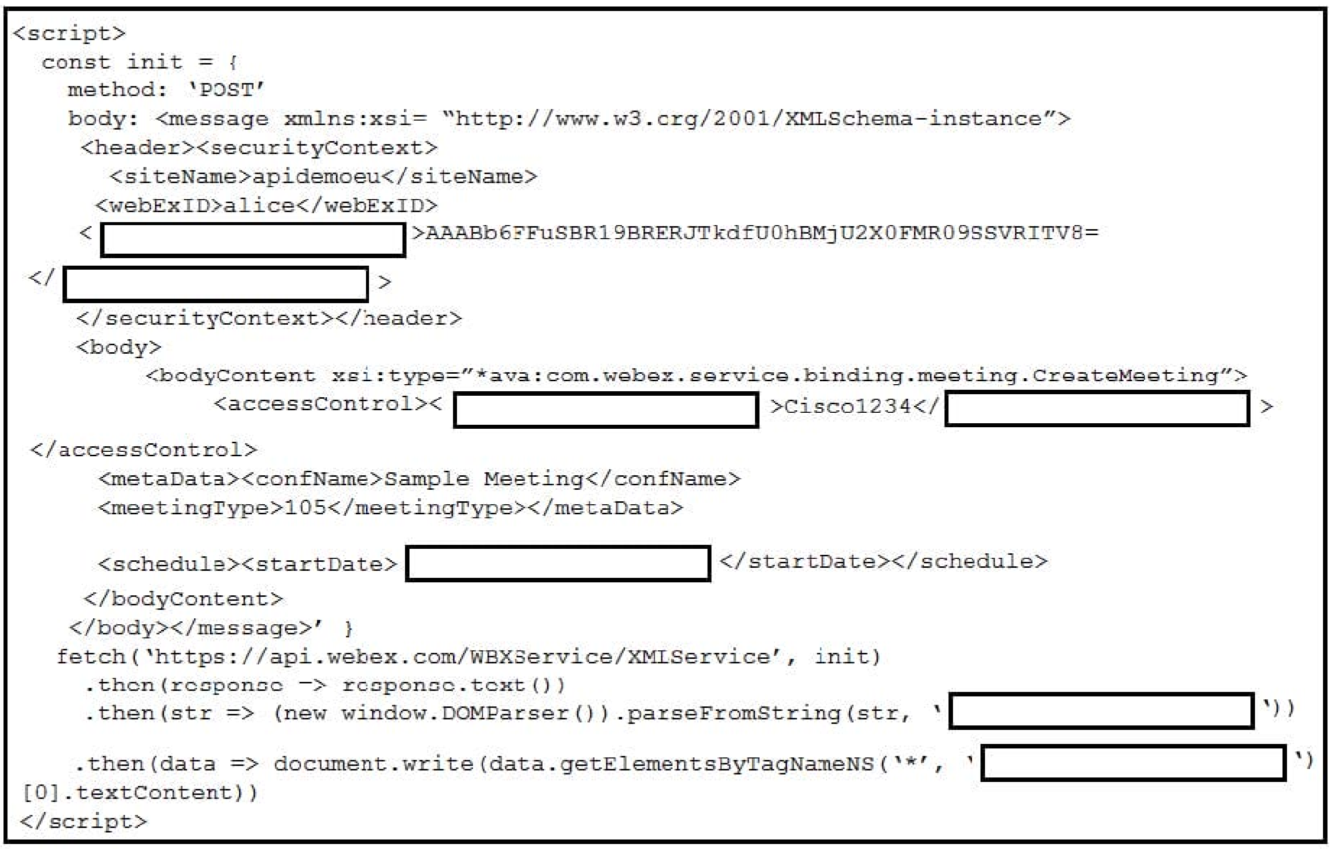
The password, although not that good, has a capital letter and numbers. Therefore, it is okay. SessionTicket credential is not expired because the error function doesn\\'t check that. We are not sure if the meeting was created successfully however, there is no wrong in the code, therefore, webexmeetings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

**QUESTION 3**

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.

Select and Place:



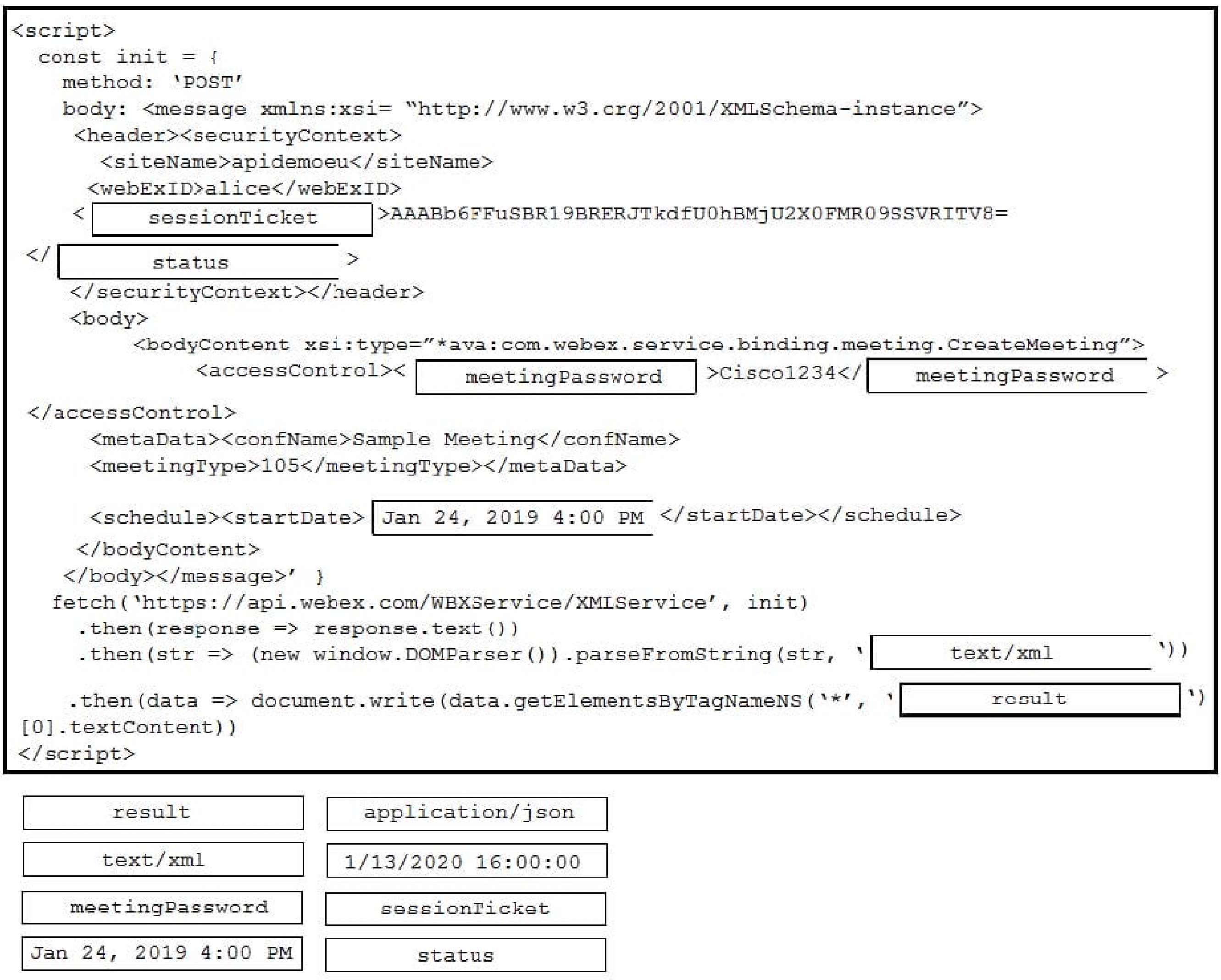
```
<script>
  const init = {
    method: 'POST'
    body: <message xmlns:xsi= "http://www.w3.crg/2001/XMLSchema-instance">
     <header><securityContext>
       <siteName>apidemoeu</siteName>
      <webExID>alice</webExID>
     <                    >AAABb6FFuSBR19BRERJTkdfU0hBMjU2X0FMR09SSVRITV8=
 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

      <schedule><startDate>                    </startDate></schedule>
     </bodyContent>
    </body></message>' }
   fetch('https://api.webex.com/WBXService/XMLService', init)
     .then(response -> response.text())
     .then(str => (new window.DOMParser()).parseFromString(str, '                    '))

    .then(data => document.write(data.getElementsByTagNameNS('*', '                    ')
[0].textContent))
</script>
```

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

Correct Answer:



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

      <schedule><startDate> 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, ' text/xml '))
    .then(data => document.write(data.getElementsByTagNameNS('*', ' 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 4**

Which element is needed to build a Web application that authenticates Webex users and can post messages under the user\\'s identity?

A. OAuth integration configured with the `messages_write\\' scope

B. bot access token

C. Guest Issuer application

D. self-signed certificate that is created from a public authority

Correct Answer: A

Reference: https://developer.webex.com/blog/real-world-walkthrough-of-building-an-oauth-webex-integration

**QUESTION 5**

Which code adds a Space Widget in an HTML script that uses the CSS Webex CDN?

A.

B.

C.

D.

Correct Answer: D

Reference: https://developer.webex.com/docs/widgets