

➤ **Vendor:** Cisco➤ **Exam Code:** 300-920➤ **Exam Name:** Developing Applications for Cisco Webex and Webex Devices (DEVWBX)➤ **New Updated Questions from** [Braindump2go](https://www.braindump2go.com) **(Updated in** [August/2020](https://www.braindump2go.com)**)****Visit Braindump2go and Download Full Version 300-920 Exam Dumps****QUESTION 50**

Drag and Drop Question

Drag and drop the code snippets onto the exhibit to create a valid Webex Meetings API request allowing Jane (an admin) to reset John's PMR pin.  
Not all options are used.

```
<?xml version="1.0" encoding="UTF-8"?>
<serv:message xmlns:serv= "http://www.webex.com/schemas/2002/06/service »
  xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
  <header>
    <securityContext>
      <siteName>apidemoeu</siteName>

      <webExID>[ ]</webExID>
      [ ]
    </securityContext>
  </header>
  <body>
    <bodyContent xsi :type= "java:com.webex.xmlapi.service.binding.user.
      [ ]">
      <webExID>[ ]</webExID>

      <personalMeetingRoom>
        <hostPIN>1337</hostPIN>
      </persomalMeetingRoom>
    </bodyContent>
  </body>
</serv:message>
```

janedoe

johndoe

SetUser

UpdateUser

<token>AAABb5HuHWUAABUY</token>

<sessionTicket>AAABb5HuHWUAABUY</sessionTicket>

**Answer:**

```
<?xml version="1.0" encoding="UTF-8"?>
<serv:message xmlns:serv= "http://www.webex.com/schemas/2002/06/service"
  xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
  <header>
    <securityContext>
      <siteName>apidemoeu</siteName>

      <webExID> johndoe </webExID>
      <SetUser>
      </SetUser>
    </securityContext>
  </header>
  <body>
    <bodyContent xsi:type= "java:com.webex.xmlapi.service.binding.user">
      <token>AAABb5HuHWUAABUY</token>
      <webExID> UpdateUser </webExID>
      <personalMeetingRoom>
        <hostPIN>1337</hostPIN>
      </personalMeetingRoom>
    </bodyContent>
  </body>
</serv:message>
```

janedoe

<sessionTicket>AAABb5HuHWUAABUY</sessionTicket>

#### QUESTION 51

Drag and Drop Question

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>
      < >AAABb6FFuSBRI9BRERJTKdfU0hBMjU2X0FMR09SSVRITV8=
    </ >
    </securityContext></header>
    <body>
      <bodyContent xsi:type="java: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/WEBService/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

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>
      <sessionTicket><input type="text" value="AAABb6FFuSBR19BRERJTkdfU0hBMjU2X0FMR09SSVRITV8=" />
    </securityContext></header>
    <body>
      <bodyContent xsi:type="*ava:com.webex.service.binding.meeting.CreateMeeting">
        <accessControl><input type="text" value="meetingPassword" /><input type="text" value="meetingPassword" />
      </accessControl>
      <metaData><confName>Sample Meeting</confName>
      <meetingType>i05</meetingType></metaData>
      <schedule><startDate><input type="text" value="Jan 24, 2019 4:00 PM" /></startDate></schedule>
    </bodyContent>
  </body></message>'
}
fetch('https://api.webex.com/WEBXService/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 52

Drag and Drop Question

Drag and drop the expressions from the right onto the Webex JavaScript SDK code on the left to create a Webex Teams space, add a person to the space, and send a welcome message. Not all expressions are used.

```

const Webex = require('webex');
const webex = Webex.init({
  credentials: {
    access_token: 'process.env.webextoken'
  }
});
webex. .create({title: 'Q3 Budget Planning'})
.then(function(x) {
  return webex. .create({
    personEmail: 'jle@devnet.i0',
    roomId: .id
  });
})
.then(function(y) {
  webex. .create({
    text: 'You've been assigned a new task!',
    roomId: y.roomId
  });
})

```

memberships

teams

rooms

messages

X

Answer:

```

const Webex = require('webex');
const webex = Webex.init({
  credentials: {
    access_token: 'process.env.webextoken'
  }
});
webex. teams .create({title: 'Q3 Budget Planning'})
.then(function(x) {
  return webex. memberships .create({
    personEmail: 'jle@devnet.i0',
    roomId: rooms .id
  });
})
.then(function(y) {
  webex. messages .create({
    text: 'You've been assigned a new task!',
    roomId: y.roomId
  });
})

```

X

#### QUESTION 53

Drag and Drop Question

Drag and drop the code to complete the JavaScript to display the first URL from a user's list of Webex Meetings recordings. Not all options are used.



```
const init = { ,
body: '<message xmlns:xsi= "http://www.v3.org/2001/XMLSchema-instance">
  <header><securityContext>
    <siteName>apidemoeu</siteName>
    <webExID>alice</webExID>
    <sessionTicket>AAABb6CkTrgAABUYAR2AD2X0FMR09SSVRITV8=</sessionTicket>
  </securityContext></header>
  <body>
    <bodyContent xsi:type="  ">

    <createTimeScope>
      <createTimeStart>11/21/2019 0:0:0</createTimeStart>
      <createTimeEnd>11/21/2019 23:59:59</createTimeEnd>
    </createTimeScope>
  </bodyContent></body></message>  };
fetch('  ',init)
  .then(response => response.text())
  .then(str => (new window.DOMParser()).parseFromString(str, 'text/xml'))
  .then(data => document.write(data.getElementsByTagName(
    '  ',
    'fileURL') [0].textContent));
```

http://www.webex.com/schemas/2002/06/service/meeting

method: 'POST'

java:com.webex.service.binding.ep.LstRecording

java:com.webex.service.binding.meeting.Recordings

http://www.webex.com/schemas/2002/06/service/ep

method: 'GET'

https://api.webex.com/WBX/Service/XMLService

https://api.webex.com/v1/recordings

Answer:

```
const init = {  method: 'POST' ,
body: '<message xmlns:xsi= "http://www.v3.org/2001/XMLSchema-instance">
  <header><securityContext>
    <siteName>apidemoeu</siteName>
    <webExID>alice</webExID>
    <sessionTicket>AAABb6CkTrgAABUYAR2AD2X0FMR09SSVRITV8=</sessionTicket>
  </securityContext></header>
  <body>
    <bodyContent xsi:type="  java:com.webex.service.binding
    .ep.LstRecording  ">

    <createTimeScope>
      <createTimeStart>11/21/2019 0:0:0</createTimeStart>
      <createTimeEnd>11/21/2019 23:59:59</createTimeEnd>
    </createTimeScope>
  </bodyContent></body></message>  };
fetch('  https://api.webex.com/WBX
Service/XMLService  ',init)
  .then(response => response.text())
  .then(str => (new window.DOMParser()).parseFromString(str, 'text/xml'))
  .then(data => document.write(data.getElementsByTagName(
    '  http://www.webex.com/schemas/
    2002/06/service/meeting  ',
    'fileURL') [0].textContent));
```

java:com.webex.service.binding.meeting.Recordings

http://www.webex.com/schemas/2002/06/service/ep

method: 'GET'

https://api.webex.com/v1/recordings

#### QUESTION 54

Drag and Drop Question

Drag and drop the code to create a valid Guest Issuer JWT token to use with the Webex Teams Browser SDK. Not all options are used.

[300-920 Exam Dumps](#) [300-920 Exam Questions](#) [300-920 PDF Dumps](#) [300-920 VCE Dumps](#)

<https://www.braindump2go.com/300-920.html>

```
const jwt = require('jsonwebtoken')
let token = jwt.sign(
  {
    'sub': ,
    'name': ,
    'iss': ,
  },
  Buffer.from(, 'base64'),
  { expiresIn:  }
);
```







Answer:

```
const jwt = require('jsonwebtoken')
let token = jwt.sign(
  {
    'sub': ,
    'name': ,
    'iss': ,
  },
  Buffer.from(, 'base64'),
  { expiresIn:  }
);
```

**Explanation:**

<https://developer.webex.com/docs/guest-issuer>

#### QUESTION 55

Drag and Drop Question

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:

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.

**Explanation:**

<https://developer.webex.com/docs/api/basics>

**QUESTION 56**

Drag and Drop Question

Refer to the exhibit. Drag and drop the code on the snippet to add the user with email "alice@example.com" as moderator of the space with the identifier "987654321".

**Create a Membership**

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

**POST** /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

```
curl -s -X [ ] "https://api.ciscospark.com/v1/[ ]" \
-H "Authorization: Bearer MDMYzTM1YTA_Pf84_1eb65fdf-ad72cae0e10f" \
-H "Content-Type: application/json" \
-d '{"[ ]": "987654321", "[ ]": "[ ]", "isModerator": "[ ]"}'
```

[personEmail] [true] [POST] [memberships] [roomId]

**Answer:**

```
curl -s -X [POST] "https://api.ciscospark.com/v1/[memberships]" \
-H "Authorization: Bearer MDMYzTM1YTA_Pf84_1eb65fdf-ad72cae0e10f" \
-H "Content-Type: application/json" \
-d '{"[roomId]": "987654321", "[personEmail]": "[alice@example.com]", "isModerator": "[true]"}'
```

**Explanation:**

<https://developer.webex.com/docs/api/v1/memberships/create-a-membership>

**QUESTION 57**

Drag and Drop Question

Drag and drop the methods from the left into the correct order of execution on the right to use webex-js-sdk in a browser to call and share the screen with another Webex user. Not all methods are used.

[300-920 Exam Dumps](#) [300-920 Exam Questions](#) [300-920 PDF Dumps](#) [300-920 VCE Dumps](#)

<https://www.braindump2go.com/300-920.html>



meeting.addMedia()	step 1
meeting.updateShare()	step 2
webex.call()	step 3
call.getMediaStreams()	step 4
webex.meetings.create()	
meeting.getMediaStreams()	

Answer:

	webex.meetings.create()
	meeting.getMediaStreams()
webex.call()	meeting.addMedia()
call.getMediaStreams()	meeting.updateShare()

**Explanation:**

[https://github.com/webex/webex-js-sdk/blob/master/packages/node\\_modules/%40webex/plugin-meetings/README.md](https://github.com/webex/webex-js-sdk/blob/master/packages/node_modules/%40webex/plugin-meetings/README.md) (see start wireless share)

**QUESTION 58**

Drag and Drop Question

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

call\_1

Widget:Call

**Explanation:**

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