



Free Questions for 300-920 by [braindumpscollection](#)

Shared by [Kennedy](#) on 24-05-2024

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

Refer to the exhibit.

```
<xsd:complexType name= "lstRecordingResponse">
  <xsd:complexContent>
    <xsd:extension base= "serv:bodyContentType">
      <xsd:sequence>
        <xsd:element name= "matchingRecords" type=
"serv:matchingRecordsType" minOccurs="0"/>
        <xsd:element name= "recording" type= "ep:recordingType" minOccurs=
"0" maxOccurs= "unbounded"/>
      ...
    <xsd:complexType name= "recordingType">
      <xsd:sequence>
        <xsd:element name= "recordingID" type= "xsd:int"/>
        <xsd:element name= "hostWebExID" type= "xsd:string"/>
        <xsd:element name= "name" type= "xsd:string"/>
        <xsd:element name= "createTime" type= "xsd:string"/>
        <xsd:element name= "streamURL" type= "xsd:string"/>
        <xsd:element name= "fileURL" type= "xsd:string"/>
        <xsd:element name= "password" type= "xsd:string" minOccurs= "0" />
      
```

A snippet from the XSD schema of the Webex Meeting XML API 'LstRecordingResponse' element is listed in the exhibit. Assuming that a variable named 'resp' exists that contains the XML response from a successful 'LstRecording' request, which code snippet correctly generates a simple report that lists meeting names and recording file download links?

- A.

```
list = (new window.DOMParser()).parseFromString(resp, 'application/json')
for (const item of lst.getElementsByTagNameNS('*', 'matchingRecords')){
document.writeln(
  <p> Meeting Name:${item.getElementsByTagNameNS('*', 'name') [0].textContent}<br>');
document.write(
  'Download:${item.getElementsByTagNameNS('*', 'fileURL') [0].textContent}<br>');
```
- B.

```
list = (new window.DOMParser()).parseFromString(resp, 'LstRecordingResponse')
for (const item of lst.getElementsByTagNameNS('*', 'matchingRecords')){
document.writeln(
  <p> Meeting Name:${item.getElementsByTagNameNS('*', 'name') [0].textContent}<br>');
document.write(
  'Download:${item.getElementsByTagNameNS('*', 'fileURL') [0].textContent}<br>');
```
- C.

```
list = (new window.DOMParser()).parseFromString(resp, 'text/xml')
for (const item of lst.getElementsByTagNameNS('*', 'matchingRecords')){
document.writeln(
  <p> Meeting Name:${item.getElementsByTagNameNS('*', 'hostWebExID') [0] }<br>');
document.write(
  'Download:${item.getElementsByTagNameNS('*', 'streamURL') [0].textContent}<br>');
```
- D.

```
list = (new window.DOMParser()).parseFromString(resp, 'text/xml')
for (const item of lst.getElementsByTagNameNS('*', 'recording')){
document.writeln(
  <p> Meeting Name:${item.getElementsByTagNameNS('*', 'name') [0].textContent}<br>');
document.write(
  'Download:${item.getElementsByTagNameNS('*', 'fileURL') [0].textContent}<br>');
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A

Question 2

Question Type: MultipleChoice

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

Options:

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

Explanation:

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

Question 3

Question Type: MultipleChoice

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?

Options:

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

Answer:

A

Explanation:

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

Question 4

Question Type: MultipleChoice

Refer to the exhibit.

```
1 const xapi = require('xapi');
2 let payload = JSON.stringify({'destination': 'new'});
3   xapi.command('HttpClient Port', {
4     Header: "Content-Type: application/json",
5     Url: "https://192.168.0.2/v1/motion",
6   },
7   payload)
8   .then((result) => {
i 9     console.log(result)
i 10  })
```

Severity ▾

Enter filter

Show history

01:19:35 Untitled > Loading...

01:19:36 [system] > Starting macros...

01:19:36 [system] > Macros ready.

```
01:19:36 Untitled > 'Unhandled promise rejection' ( code: 0,
    message: 'HttpClientPostResult',
    data:
      ( status: 'Error',
        Message: 'SSL peer certificate or SSH remote
key was not OK' } }
```

What causes the error message?

Options:

- A- xapi must be enabled for promises.
- B- HttpClient AllowInsecureHTTPS has not been enabled.
- C- The NODE_TLS_REJECT_UNAUTHORIZED environment variable must be set to 0.
- D- HttpClient must be changes to HttpClient.

Answer:

B

Explanation:

[https://help.webex.com/en-us/nthg6le/Sending-HTTP-Requests-from-a-Board-Room-or-Desk- Device](https://help.webex.com/en-us/nthg6le/Sending-HTTP-Requests-from-a-Board-Room-or-Desk-Device)

Question 5

Question Type: MultipleChoice

Refer to the exhibit.

```
1 const webex = windows.webex = Webex.init();
2
3 webex.once('ready' , () => {
4   const jwt = document.getElementById('context').value;
5
6   webex.authorization.requestAccessTokenFromJwt({ jwt })
7     .then(() => {
8       if (webex.canAuthorize) {
9         alert('Authenticated!');
10      }
11    })
12    .catch((err) => {
13      alert('Authentication.error:' +err);
14    });
15});
```

On line 4, the script retrieves a context from a DOM element that was generated from a server-side component. How does that server-side component obtain the value for the 'context' element?

Options:

- A- by opening a dialog asking the end-user to paste his personal access token
- B- by completing an authorization code grant flow using the identifier and secret of an OAuth integration
- C- by embedding the access token of a Bot account
- D- by creating a guest token using the identifier and secret of a Guest Issuer application

Answer:

B

Question 6

Question Type: DragDrop

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">
```

Answer:

```
<header>
  <securityContext>
    <siteName>apidemoeu</siteName>
  </securityContext>
  <webExID> [redacted] </webExID>
```

Question 7

Question Type: Multiple Choice

```
</bodyContent xsi :type= "java:com.webex.xmlapi.service.binding.user.
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?

<token>AAABb5HuHWUAABUY</token> >

<sessionTicket>AAABb5HuHWUAABUY</sessionTicket> onTicket>

Options:

A- 401

B- 403

C- 429

D- 501

Answer:

C

Explanation:

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

Question 8

Question Type: MultipleChoice

Refer to the exhibit.

```
const webex = require('webex/env');
const BOT_ID = 'Y2ABCDEFG3VzL1BFT1BMRS9mNWTy1jGIyZilm';
// ...
let notification = {
  "id": "Y21zY29zcGFyazovL3VzL1dFQkwMQ50DgxNjA5NDk3",
  "name": "notifier",
  "resource": "messages",
  "event": "created",
  "filter": "roomId=Y21zy29zcGFyazovL3VzL1Jyju4LtkxNDctZjE0YmIwYzRkMTU0",
  "orgId": "OTZhYmMyYWEtM2RjYy0xMWUuzNDgxOWNkYzlh",
  "createdBy": "Y21zY29zcGFyazovL3VzLOFQUExOTdjODUwM2YyNjZhYmY2NmMSOTIIYzFm",
  "ownedBy": "creator",
  "status": "active",
  "actorId": "Y21zY29zcGFyazovL3VzL1BFT18MRS9mNWIyijGIyZilmOWMONDdmMjkwNDY",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzLZlYjAtmzRkNC0xMWVhLTgwZmMtYjFlNmRiMjk0YjM5",
    "roomId": "Y21zY29zcGFyazovL3VzLiJPT00vYmJjZWZyju4LtkxNDctZjE0YmIwYzRkMTU0",
    "personId": "Y21zY29zcGFyazovL3VzL1BFTijctOGIyZilmOWMONDdnMjkwNDY",
    "personEmail": "alice@example.com",
    "created": "2019-10-18T14:26:26.000Z"
  }
};
```

A webhook has been created so that an application is notified when users mention a bot in a Webex Teams space. The exhibit shows an example of a notification received by the application. Which code snippet correctly processes the JSON payload using the Webex Node.js SDK in order to print out messages that mention the bot?

A.

```
if ((notification.name == 'notifier') &&
    (notification.event == 'created')) {
  if (notification.resource == 'messages') {
    webex.messages.get(notification.data.id).then(
      msg => console.log(msg.text));
  }
}
```

B.

```
if ((notification.resource == 'messages') &&
    (notification.event == 'created')) {
  webex.messages.get(notification.data).then(
    msg => console.log(msg.text));
}
```

C.

```
if ((notification.resource == 'messages') &&
    (notification.event == 'created')) {
  if (notification.data.personId !== BOT_ID) {
    webex.messages.get(notification.data.id).then(
      msg => console.log(msg.text));
  }
}
```

D.

```
if ((notification.resource == 'notifier') &&
    (notification.event == 'created')) {
  if (notification.data.personId !== BOT_ID) {
    webex.messages.get(notification.data.id).then(
      msg => console.log(msg.text));
  }
}
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 9

Question Type: MultipleChoice

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?

Options:

- 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 10

Question Type: MultipleChoice

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

Options:

- 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

To Get Premium Files for 300-920 Visit

<https://www.p2pexams.com/products/300-920>

For More Free Questions Visit

<https://www.p2pexams.com/cisco/pdf/300-920>

