

100% Money Back
Guarantee

Vendor:Cisco

Exam Code:300-920

Exam Name:Developing Applications for Cisco Webex
and Webex Devices (DEVWBX)

Version:Demo

QUESTION 1

```
1  const WebExClient = require('webex-api-client');
2  const securityContext = {
3    webExID: 'Test User',
4    password: 'pass123',
5    siteName: 'hello-world'
6  };
7
8  const requestBuilder = new WebExClient.Builder
9  (securityContext, 'https://hello-world.webex.com/WBXService/XMLService');
10
11 const createMeeting =
12   requestBuilder
13   .metaData({
14     confName: 'Sample Meeting',
15     meetingType: 1,
16   })
17   .participants({
18     attendees: [
19       {
20         name: 'Jane Doe',
21         email: 'jdoe@gmail.com'
22       }
23     ]
24   })
25   .schedule({
26     startDate: new Date(),
27     _____,
28     _____
29   })
30   .setService('CreateMeeting')
31   .build();
32
33 createMeeting
34   .exec()
35   .then((resp) => console.log('success'));
36
37
```

Refer to the exhibit. Which code for blank lines 26, 27, and 28 gives invitees 900 seconds before the scheduled time to join the meeting, sets the meeting to last for 30 minutes, and sets the meeting timezone to Pacific US?

A.

```
26 openMeeting: 900
27 stopTime: 30,
28 timeZone: 4
```

B.

```
26 openTime: 900
27 duration: 30,
28 timeZoneID: 4
```

C.

```
26 joinTime: 900
27 meetingLength: 30,
28 timezone: 5
```

D.

```
26 joinTime: 900
27 endTime: 30,
28 timeZoneID: pt_us
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

Reference: <https://github.com/cisco-ie/webex-api-client>

QUESTION 2

```
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});
```

Refer to the exhibit. 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?

- 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

Correct Answer: B

QUESTION 3

```

const webex = require('webex/env');
const BOT_ID = 'Y2ABCDEFG3VzL1BFT1BMRS9mNWTy1jGIyZilm';
// ...
let notification = {
  "id": "Y21zY29zcGFyazovL3VzL1dFQkwMQ50DgxNjA5NDk3",
  "name": "notifier",
  "resource": "messages",
  "event": "created",
  "filter": "roomId=Y21zy29zcGFyazovL3VzL1Jyju4LtkxNDctZjE0YmIwYzRkMTU0",
  "orgId": "OTZhYmMyWEtM2RjYy0xMWUuzNDgxOWNkYz1h",
  "createdBy": "Y21zY29zcGFyazovL3VzL0FQUEXOTdjODUwM2YyNjZhYmY2NmMSOTIiYzFm",
  "ownedBy": "creator",
  "status": "active",
  "actorId": "Y21zY29zcGFyazovL3VzL1BFT18MRS9mNWIyijGIyZilmOWMONDdmMjkwNDY",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzLz1YjAtMzRkNC0xMWVhLTgwZmMtYjFlNmRiMjk0YjM5",
    "roomId": "Y21zY29zcGFyazovL3VzLiJPT00vYmJjZWZyju4LtkxNDctZjE0YmIwYzRkMTU0",
    "personId": "Y21zY29zcGFyazovL3VzL1BFTijctOGIyZilmOWMONDdmMjkwNDY",
    "personEmail": "alice@example.com",
    "created": "2019-10-18T14:26:26.000Z"
  }
};

```

Refer to the exhibit. 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));
  }
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

QUESTION 4


```

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 5

```
{
  "items":
  {
    "id": "Y21zY29zcGFyazovL3VzL1LWT4Y2UtMTEzZjhhZmMxNGJl",
    "title": "DevNet Dev Support Questions",
    "lastActivity" : "2019-12-13T21:32:11.424Z",
    "created": "2016-01-22T19:07:24.785Z",
    ...
  },
  {
    "id": "Y21zY29zcGFyazovL3VzWU1LWT0ZjQtZmJmMjI3Y2ZmYWYz",
    "title": "DevNet Program Questions",
    "lastActivity" : "2019-12-13T16:53:14.920Z",
    "created": "2016-02-24T18:52:35.013Z",
    ...
  }
]
}
```

Refer to the exhibit. Which Webex Teams REST API request has generated the response body in the exhibit?

- A. GET/v1/rooms?sortBy=lastactivity
- B. GET/v1/rooms?sortBy=created
- C. GET/v1/rooms?max=1
- D. GET/v1/spaces?orderBy=lastActivity

Correct Answer: C

Reference: <https://developer.webex.com/docs/api/v1/rooms/list-rooms>

QUESTION 6

DRAG DROP

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

Select and Place:


```
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');
```

Correct 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 7

Which two statements about Webex Teams refresh tokens are true? (Choose two.)

- A. The refresh token is useless without the client ID and client secret.
- B. An attacker can use the refresh token to send messages on behalf of the user.
- C. The refresh token is used to generate a new access token.
- D. A new refresh token cannot be granted until the client ID is invalidated.
- E. The refresh token does not expire.

Correct Answer: BC

Reference: <https://auth0.com/learn/refresh-tokens/>

QUESTION 8

```
xapi.on('ready', () => {
  console.log("connection successful");

  // Start a call

  * _____ *

  .then((call) => {
    console.log('Started call with status: ${call.status}, id: ${call.CallId}');
  });
});
```

Refer to the exhibit. 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

Correct Answer: D

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

QUESTION 9

DRAG DROP

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.

Select and Place:

```

const init = { [redacted] },
body: '<message xmlns:xsi= "http://www.v3.org/2001/XMLSchema-instance">
  <header><securityContext>
    <siteName>apidemoeu</siteName>
    <webExID>alice</webExID>
    <sessionTicket>AAABb6CkTRgAABUYARZAD2X0FMR09SSVRITV8=</sessionTicket>
  </securityContext></header>
  <body>
    <bodyContent xsi:type=" [redacted] ">

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

Correct 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>AAABb6CkTRgAABUYARZAD2X0FMR09SSVRITV8=</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.getElementsByTagNameNS(
    '
      http://www.webex.com/schemas/
      2002/06/service/ep
    ',
    'fileURL' ) [0].textContent));

```

```

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

```

```

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

```

```

method: 'GET'

```

```

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

```

QUESTION 10

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.

Correct Answer: A

Reference: <https://pdfslide.net/documents/webex-we.html> (p.216)

QUESTION 11

DRAG DROP

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.

Select and Place:

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

- | | |
|---|---|
| <input type="text" value="'Time span or seconds count'"/> | <input type="text" value="'Guest Application Identifier'"/> |
| <input type="text" value="'User Display Name'"/> | <input type="text" value="'Guest User Key'"/> |
| <input type="text" value="'Guest Application Secret'"/> | <input type="text" value="'User Identifier'"/> |

Correct Answer:


```
const jwt = require('jsonwebtoken')
let token = jwt.sign(
  {
    'sub': 'User Identifier' ,
    'name': 'User Display Name' ,
    'iss': 'Guest Application Identifier' ,
  },
  Buffer.from('Guest Application Secret', 'base64'),
  { expiresIn: 'Time span or seconds count' }
);
```

'Guest User Key'

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

QUESTION 12

Which user role implicitly has permission in the Events API to retrieve "message created" events from other users in the same organization?

- A. Compliance Officer
- B. User and Device Administrator
- C. Full Administrator
- D. User Administrator

Correct Answer: A

Reference: <https://developer.webex.com/docs/api/guides/compliance>