

➤ **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** [Dec./2020](#)**)****Visit Braindump2go and Download Full Version 300-920 Exam Dumps****QUESTION 18**

Refer to the exhibit. A developer must construct an HTTP Request to use the XML API to set a Personal Meeting Room PIN for a given user.

Which code completes the code to create the request?

```
const xml = `<?xml version= "1.0" encoding= "UTF-8"?>
<serv:message xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
xmlns:serv="http://www.webex.com/schemas/2002/06/service"
xsi:schemaLocation="http://www.webex.com/schemas/2002/06/service"
http://www.webex.com/schemas/2002/06/service/service.xsd">
  <header>
    <securityContext>
      <webExID>admin@cisco.com</webExID>
      <password>password</password>
      <siteName>cisco</siteName>
      <returnAdditionalInfo>true</returnAdditionalInfo>
    </securityContext>
  </header>
  <body>
    <bodyContent xsi:type= "java:com.webex.service.binding.user.SetUser">
      <webExId>user@cisco.com</webExId>
      <personalMeetingRoom>
        <hostPIN>3421</hostPIN>
      </personalMeetingRoom>
    </bodyContent>
  </body>
</serv:message>;
var xmlhttp = new XMLHttpRequest();

<< missing code >>

xmlhttp.setRequestHeader('Content-Type', 'text/xml');
xmlhttp.send(xml);
```

- A. xmlhttp.open("GET", "https://cisco.webex.com/WBXService/XMLService");
- B. xmlhttp.open("PATCH", "https://cisco.webex.com/WBXService/XMLService");
- C. xmlhttp.open("PUT", "https://cisco.webex.com/WBXService/XMLService");
- D. xmlhttp.open("POST", "https://cisco.webex.com/WBXService/XMLService");

Answer: D

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

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

Explanation:

The post method can be used for HTTP request that sets up a personal meeting room PIN for a user.

QUESTION 19

Which expression is a valid Webex Teams webhook filter?

- A. personEmail=person@example.com+roomId=abc123
- B. personEmail=person@example.com-roomId=abc123
- C. personEmail=person@example.com&roomId=abc123
- D. personEmail=person@example.com,roomId=abc123

Answer: C**Explanation:**

You can also use more than one filter in a webhook. To use multiple filters, combine them with the "&" symbol. For example, to create a webhook that only sends notifications when a specific person performs an action in a specific room, such as sending a message or creating a membership, combine the personEmail and roomId filters.
<https://developer.webex.com/docs/api/guides/webhooks>

QUESTION 20

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);
});
```

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 21

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?

```
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    })
31    .setService('CreateMeeting')
32    .build();
33
34  createMeeting
35    .exec()
36    .then((resp) => console.log('success'));
37
```

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

Answer: B

Explanation:

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

QUESTION 22

Which two items are needed to give a Webex user the ability to archive all Webex Teams messages for an organization? (Choose two.)

- A. Give the user "Read-only administrator privileges" in the Webex Control Hub.
- B. Create an Integration app with all "spark_compliance" read scopes enabled.
- C. Use the Webex Meetings XML API "SetUser" to update the user's "<roSiteAdmin>" value to "TRUE".
- D. Configure the user as a "Compliance Officer" in the Webex Control Hub.
- E. Create a Bot app with all "webex_compliance" read scopes enabled.

Answer: AD

Explanation:

https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/cloudCollaboration/spark/esp/Webex-Teams-Security-Frequently-Asked-Questions.pdf

QUESTION 23

One of the challenges of online training is measuring class effectiveness. Which two tools does Cisco WebEx Training Center use to measure this? (Choose two.)

- A. visual attention indicator
- B. polling
- C. self-check quizzes
- D. application sharing

Answer: AB

QUESTION 24

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

Explanation:

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

QUESTION 25

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?

```
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>Cisc01234</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!') } });
```

- 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 26

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?

```
const webex = require('webex/env');
const BOT_ID = 'Y2ABCDEF3VzL1BFT1BMRS9mNWTy1jGIyZilm';
// ...
let notification = {
  "id": "Y21zY29zcGFyazovL3VzL1dFQkwMQ50DgxNjA5NDk3",
  "name": "notifier",
  "resource": "messages",
  "event": "created",
  "filter": "roomId=Y21zy29zcGFyazovL3VzL1JyYjU4LtkxNDctZjE0YmIwYzRkMTU0",
  "orgId": "OTZhYmMyYWEtM2RjYy0xMWUzNDgxOWNkYzlh",
  "createdBy": "Y21zY29zcGFyazovL3VzL0FQUEXOTdjdjODUwM2YyNjZhYmY2NmMSOTIiYzFm",
  "ownedBy": "creator",
  "status": "active",
  "actorId": "Y21zY29zcGFyazovL3VzL1BFT18MRS9mNWIyijGIyZilmOWMONDdmMjkwNDY",
  "data": {
    "id": "Y21zY29zcGFyazovL3VzL1ZlYjAtMzRkNC0xMWVhLTgwZmMtYjFlNmRiMjk0YjM5",
    "roomId": "Y21zY29zcGFyazovL3VzL1JPT00vYmJjZWZyYjU4LtkxNDctZjE0YmIwYzRkMTU0",
    "personId": "Y21zY29zcGFyazovL3VzL1BFT1ijctOGIyZilmOWMONDdmMjkwNDY",
    "personEmail": "alice@example.com",
    "created": "2019-10-18T14:26:26.000Z"
  }
};
```

- 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));
  }
}
```

Answer: B

QUESTION 27

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

```
{
  "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"
}
```

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

Answer: C

Explanation:

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

QUESTION 28

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?

```
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 });
```

- 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