



"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

[www.atmicnetworks .com](http://www.atmicnetworks.com)

Warning: Keep connected with our support team
for latest updates

Question:1

DRAG DROP

Drag and drop the components to create the authentication header used for the Webex Teams API.

Not all options are used.

admin:Password 12345!

Authorization

Token

Bearer

BH782kjh-2jkhmn63rlkj3lkjnm3klj3

Authentication

Basic

Answer:

Authorization

Bearer

Reference:

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

Question:2

Refer to the exhibit.

```
const xml - '<?xml version= "1.0" encodings "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>admin9cisco.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>342K/hostPIN>
        </personalMeetingRoom>
      </bodyContent>
```

```
</body>
</serv:message>
var xmlhttp = new XMLHttpRequest();

« missing code »

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

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?

- 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

Explanation:

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

Question:3

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.

Reference:

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

Question:4

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?

```
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);
});
```

```
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);
});
```

A. OptionA B. OptionB C. Option C D. Option D

Answer: A**Explanation:**

The qs: option is required to list all roomkit devices. Product: 'RoomKit' is the correct option because it will list all roomkitdevices.

Question:5**DRAG DROP**

Drag and drop the code onto the snippet to construct the JavaScript to create a new meeting with the Webex Meetings XML API. Options can be used more than once.

```
var http = require('https');
var xml = '<?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>mySite</siteName>
<webExID>Ciscouser</webExID>
```

```

    <[redacted]>
    PD4wiqNfRkxBR19B...RERJTkdfU0hBMjU2X0FMR09SSVRITV8=
    </[redacted]>
    </securityContext></header>
    <body>
    <bodyContent xsi:type="java.com.webex.service.binding.meeting.CreateMeeting">
    <metaData><confName>Sample Meeting</confName>
    <meetingType>105</meetingTypeX/metaData>
    <scheduleXstartDate>12/13/2019 11:59:5</startDate></schedule>
    </bodyContent>
    </body>
    </serv:message>
    var req = http.request('https://api.webex.com/WBXService/XMLService',

    );
    req.write(xml);
    req.end();

```

```
{ 'method' : 'POST' }
```

```
{ 'method' : 'PUT' }
```

```
accessToken
```

```
sessionTicket
```

Answer:

```

var http = require('https');
var xml = '<?xml versions "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>mySite</siteName>
<webExID>Ciscouser</webExID>
< (method' : 'POST') >

PD4wiqNfRkxBR19B...RERJTkdfU0hBMjU2X0FMR09SSVRITV8=
'accessToken.>
</securityContext></header>
<body>
<bodyContent xsi:type="java.com.webex.service.binding.meeting.CreateMeeting">
<metaDataXconfName>Sample Meeting</confName>
<meetingType>105</meetingTypeX/metaData>
<schedule><startDate>12/13/2019 11:59: 5</startDate></schedule>
</bodyContent>
</body>
</serv:message>
var req = http.request('https://api.webex.com/WBXService/XMLService',
    sessionTicket );
req.write(xml);
req.end();

```

Question:6

Refer to the exhibit.

```
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: 'jdoe3gmail.com'
22          )
23        ]
24      ))
25      .schedule((
26        startDate: new Date(),
27
28
29
30      ))
31      .setService('CreateMeeting')
32      .build();
33  createMeeting
34    .exec()
35    .then((resp) => console.log('success'));
36
37
```

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. OptionA B. OptionB C. OptionC D. Option D

Answer: B

Reference:

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

Question:7

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

Reference:

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

Question:s.pdf

Question: 8

DRAG DROP

Drag and drop the code to complete the JavaScript snippet so that it: retrieves the details of an individual user checks what licenses they have already updates their account with a new license Options can be used more than once.

```
const request = require("request");
var adminToken = "VALID_ADMIN_TOKEN";
var personId = "VALID_PERSON_ID";
var licenseId = "VALID_LICENSE_ID";
function buildOptions(sendURL) { return (
  url: sendURL, headers: { 'Content-Type': 'application/json', 'Authorization': 'Bearer $ (adminToken)' }
}

var peopleOptions = buildOptions('https://api.ciscospark.com/v1/ | _____ | /${personId}')
/${personId}');
request.get(peopleOptions, function(error, response) { var json = JSON.parse(response.body);

if (json. | _____ | .indexOf ( | _____ | ) < 0) {

json. | _____ | .push( | _____ | );

peopleOptions["body"] = JSON.stringify(json) : request.put(peopleOptions, function(error, response) { var json = JSON.parse(response.body) ; console.log (json) ; } ) ;
} } );

licenseId person

people licenses
```

Answer:

```

var adminToken = 'VALID_ADMIN_TOKEN';
var personId = 'VALID_PERSON_ID';
var licenseId = 'VALID_LICENSE_ID';
function buildOptions(sendURL) {
    return (
        url: sendURL, headers: {
            'Content-Type': 'application/json', 'Authorization': 'Bearer $ (adminToken)
        }
    );
}

var peopleOptions = buildOptions('https://api.ciscospark.com/v1/
/S/{personId}');
request.get(peopleOptions, function(error, response) { var json =
JSON.parse(response.body);

    licenses
        .indexOf(licenseId) > -1 ? {

json.
    people
        .push(
            person
            peopleOptions['body'] = JSON.stringify(json) : request.put(peopleOptions, function(error, response) { var json = JSON.parse(response.body) ;
            console.log(json) ;
        }
    );
}

```

Question:9

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

Question: 10

Refer to the exhibit.

```

const request = require('request')
request.post({url: 'https://api.webex.com/WBXService/XMLService', headers: {'Content-Type': 'text/xml'},
body : '<messageXheaderXsecurityContext>
<siteName>apidemoeu</siteName>
<webExID>ali ce</webExID>
<sessionTicket>AAABb6DpCJgAABLnj2X0FMR09ESVRITV8=</sessionTicket>
</securityContextX/header>
<bodyXbodyContent xsi: type- "java: com. webex. service .bitiding.meeting. CreateHeeting">
<accessControl>
<meetingPasswordXlisco123 4</meetingPassword>
</accessControl>
<metadataXconfName>3ample Meeting</confName>
<meetingType>105</meetingTypeX/metadata>
<scheduleXstartDate>1/13/2020 16:00 :00</startDateX/schedule>
</bodyContentX/bodyX,message></>', function (error, response, body) {

```

```
if ({error £6 response.statusCode =2001 { console.log('Here!') } 1);
```

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 <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: 11

Refer to the exhibit.

```
const webex = require('webex/env');
const BOTID = 'Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm';
// - let notification = (
  "id": "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm",
  "name": "notifier",
  "resource": "messages", "event": "created", "filter":
  "roomId=Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "orgId":
  "OTZhYmMyYWEtM2RjYyOxMwUzNDgxOWNkYzlh", "createdBy":
  "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "ownedBy": "creator", "status":
  "active", "actorId": "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "data": {
    "id": "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "roomId":
    "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "personId":
    "Y21zY29zcGFyazovL3VzLlBFTlBMRs9mNWYlYjGlyZilm", "personEmail": "alice@example.com",
    "created": "2019-10-18T14:26:26.000Z"
```

D;

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?

```

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

```

```

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

```

```

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

Answer: B

Question: 12

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?

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

Answer: C

Reference:

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

Question: 13

DRAG DROP

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>

<personalMe etingRoom>

<hostPIN>1337</hostPIN>

</persomalMeeti ngRo om>

</bodyContent>

</body>

</serv:message>

janedoe

johndoe

SetUser

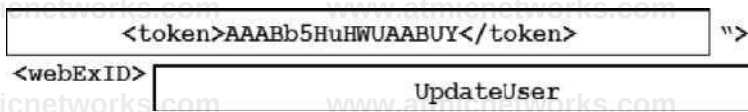
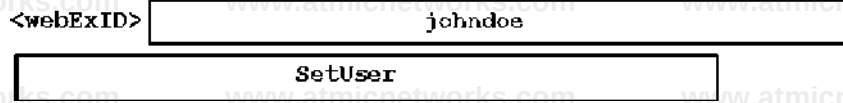
UpdateUser

<token>AAABb5HuHWUAABUY</token>

<sessionTicket>AAABb5HuHWUAABUY</sessionTicket>

Answer:

<?xml version="1.0" encoding="UTF-8"?>



```
<serv:message xmlns:serv="http://wvrw.webex.com/schemas/2002/06/service" » xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <header>
    <securityContext>
      <siteName>apidemoeu</siteName>
    </securityContext>
  </header>
  <body>
    <personalMeetingRoom>
      <hostPIN>1337</hostPIN>
    </personalMeetingRoom>
  </body>
</serv:message>
```

</webExID>

</webExID>

Question: 14

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?

- 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: 15

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) => {
9     console.log(result)
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?

- 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 changed to HttpClient.

Answer: B

Reference:

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

Device

Question: 16

Refer to the exhibit.

```
const jsxapi = require('jsxapi');
const speaker = require('./speaker')
const xapi = jsxapi.connect('ssh://10.131.202.230 username: 'admin', password:
1
2
3 let state= 0 function toggle(state) {
4   if (state = 0) {return 1} else {return 0}
5
6   xapi.on('ready', () => {
7     xapi.event on { ' ' , (x) => {
8       if (x.PanelId = 'panel_1') { state = toggle(state) if (state=1) {
9         speaker.control({"on": true}); else {
10           speaker.control({"on": false})
11
12
13
14
15
```

16 An 'Action Button' with identifier 'panel_1' is deployed to the Touch10 interface of a Room Series device.
17 Which event must be inserted into line 14 to turn the music on/off in the conference room?
18

- 19 A. UserInterface Extensions Event Pressed
- 20 B. UserInterface Extensions Panel Clicked
- 21 C. UserInterface Extensions Widget GetValue
- 22 D. UserInterface Extensions Widget Action
- 23
- 24

Answer: A

Reference:

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

Question: 17

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

- 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: AB

Reference:

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

Question: 18

Refer to the exhibit.

```
<xsd:complexType name="LstRecordingResponse">
  <xsd:complexContent>
    <xsd:extension base="serv:bodyContent" type="sequence"/>
  </xsd:complexContent>
  <xsd:sequence>
    <xsd:element name="matchingRecords" type="serv:matchingRecordsType" minOccurs="0" />
    <xsd:element name="recording" type="epireordingType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <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" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:complexType>
```

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>');
```

A. OptionA B. OptionB C. OptionC D. Option D

Answer: A

Question: 19

Which xAPI access mechanism requires separate connections for commands and notifications?

- A. Serial
- B. WebSocket
- C. HTTP/HTTPS
- D. SSH

Answer: D

Reference:

<https://github.com/CiscoDevNet/labs-xapi/blob/master/labs/collab-xapi-intro/5.md>

Question: 20

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.

<script>

```

const init = {
  method: 'POST'
  body: <message xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
    <headerXsecurityContext>
      <siteName>apidemoeu</siteName>
      <webExID>alice</webExID>
      < |_____ |>MaBb6FFuSBR19BRERJTkafrahBMjr2X(JFMRg9SS7R:TV8=

    </securityContext></header>
    <body>
      <bodyContent xsi:type="*ava:com.webex.service.binding.meeting.CreateMeeting">

        <accessControl>< |_____ 1 _____ |>
      </accessControl>
      <metaData><confName>Sample Meeting</confName>
      <meetingType>105</meetingTypeX/metaData>
      <scheduleXstartDate> |_____ | </startDateX/schednle>
    </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 j | sessionTicket |
| I Jan 24, 2015 4:00 PM | status |

Answer:

```

<script>
const init = {
  method: 'POST'
  body: <message xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance">
    <headerXsecurityContext>
      <siteName>apidemoeu</siteName>
      <webExID>alice</webExID>
      < |sessionTicket |>AAABb6FFuSBR19BRERJTkdfU0hBMjU2X0FMR09SSVRITV8=

      < |_____ status _____ |>
    </securityContextX/header>
    <body>
      <bodyContent xsi:type="*ava:com.webex.service.binding.meeting.CreateMeeting">
        <accessControl>< |meetingpassword |>Cisco1234</~ meetingpassword >
      </accessControl>
      <metaDataXconfName>Sample Meeting</confName>
      <meetingType>105</meetingTypeX/metaData>
      <scheduleXstartDate> |Jan 24, 2019 4:00 PM | </startDateX/schedule>
    </bodyContent>
  </bodyX/message>' }
fetch('https://api.webex.com/WBXService/XMLService', init)
  .then(response => response.text())
  .then(str => (new window.DOMParser()).parseFromString(str, 'text/xml' _____ p) )
  .then(data => document.write(data.getElementsByTagNameNS( '*' , '_____ result _____' )
[0].textContent)) </script>

```

Question: 21

Which two capabilities are currently supported by the Webex Meetings XML API? (Choose two.)

- A. Request a recording link for playback.
- B. Send a text message to the meeting host.
- C. Request permissions to schedule on someone else's behalf.
- D. Schedule a new meeting.
- E. Send a problem report.

Answer: CD

Reference:

<https://community.cisco.com/t5/cloud-collaboration/api-for-scheduling-a-webex-meeting-on-behalf-of-other-user/td-p/3472319>

Question: 22

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.

Answer: BC

Reference:

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

Question: 23

With CE 9.8 and above, which two statements are correct when an application is sending and receiving data over a connection established with an xAPI interface? (Choose two.)

- A. All Serial, SSH, and WebSockets can be used to send and receive data.
- B. HttpClient can be used to send requests but not receive responses.
- C. HttpFeedback is the only option to receive data.
- D. The HttpClient command can be used to send requests and read responses over HTTP.
- E. WebSockets is the only option to send and receive data.

Answer: BD

Reference:

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

Question: 24

DRAG DROP

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.io',
    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.io',
    roomId: rooms id
  });
});

webex. messages .create({
  text: 'You've been assigned a new task!',
  roomId: y.roomId
});
});

```

Question: 25

Refer to the exhibit.

```
xcommand UserInterface Extensions List
```

```
OK
```

```
*r ExtensionsListResult (status=OK) :
```

```
* r ExtensionsListResult Extensions Version: "1.5"
```

```
* r ExtensionsListResult Extensions Panel1 Icon: Lightbulb
```

```
* r ExtensionsListResult Extensions Panel1 Type: Statusbar
```

```
* r ExtensionsListResult Extensions Panel1 Name: Panel
```

```
*r ExtensionsListResult Extensions Panel 1 PanelId: "panel_1"
```

```
^r ExtensionsListResult Extensions Panel1 Order: 1
```

```
*r ExtensionsListResult Extensions Panel1 Page 1 Name: "Page"
```

```
*r ExtensionsListResult Extensions Panel1 Page 1 Row 1 Name: "Row"
```

```
* r ExtensionsListResult Extensions Panel1 Page 1 Row 1 Widget 1 WidgetTd: "widget_1"
```

```
* r ExtensionsListResult Extensions Panel1 Page 1 Row 1 Widget 1 Type: Slider
```

```
* r ExtensionsListResult Extensions Panel1 Page 1 Row 1 Widget 1 Options: "size-3"
```

```
** end
```

With this in-room configuration, which command successfully sets the slider to 50?

- A. xcommand UserInterface Extensions Widget SetSlider Value: 50
- B. xcommand UserInterface Extensions Widget Slider: 50 Widget: "widget_1"
- C. xconfiguration UserInterface Extensions widget_1 50
- D. xcommand UserInterface Extensions Widget SetValue WidgetId: "widget_1" Value: 50

Answer: A

Reference:

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

Question: 26

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

Answer: A

Reference:

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

Question: 27

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.

```
const init = {
```

```
body: '<message xmlns:xsi= "http://www.v3.org/2001/XMLSchema-instance">
  <headerXsecurityContext>
    <siteName>apidemoeu</siteName>
    <webExID>alice</webExID>
    <sessionTicket>AAABb6CkTRgAABUYARZAD2X0FMR09SSVRITV8=</sessionTicket>
  </secnrityContextX/header >
  <body>
    <bodyContent xsi:type=
    <createTimeScope>
      <createTimeStart>11/21/2019 0:0:0</createTimeStart>
      <createTimeEnd>11/21/2019 23:59:59</createTimeEnd>
    </createTimeScope>
  </bodyContent></bodyX/message> ';
fetch( init)
```

```
.then(response => response.text())
.then(str => (new window.DOMParser()).parseFromString(str, 'text/xml'))
.then(data => document.write(data.getElementsByTagNameNS(
```

```
fileURL') [0].textContent));
http://www.webex.com/schemas/ 2002/06/service/meeting
```

```
method: 'POST'
```

```
j ava: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>AAABb6CkTRgAABUYARZAD2X0FMR09SSVRITV8=</sessionTicket>
    </securityContext></header>
    <body>
      <bodyContent xsi:type="java:com.webex.service.binding
        .ep.LstRecording
      </bodyContent>
      <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/meeting
      '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

Question: 28

Which two filters are valid for limiting a webhook? (Choose two.)

- A. roomId=<roomId>
- B. personId!=<personId>
- C. spaceId=<spaceId>
- D. personId=<personId>\$spaceId=<spaceId>
- E. personId=<personId>\$roomId=<roomId>

Answer: AB

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

Refer to the exhibit.

18:42:36 exam 'macro started'

```
18:42:36 exam > 'Unhandled precise rejection' (code: 0, message:
```



Widget id

volume.slider

Volume



Import from file

Create new macro

exam

ultrasound

```

const xapi = require('xapi');

2 3v
4 3 xapi.on('ready', () => {
6v
7v
8 xapi.status.get('*') .then ((volume) -> {
9 xapi.command('* Extensions widget Se

10 I Widget Id: V i<
11 Value: volume* 255 / 100
12 n;
13v I);
14v

15 16 xapi.event. on (' Widge ', (event) => {
17 18 ((event.WidgetId = "time" ) && (event.Type "
19 20 const volume = Math.round(parseInt(event.Value) * 100/255);
xapi.command(' (Level: volume) });
return;
}

```

H;

Seventy • Biter filter [www.atmicnetworks.com](#) [www.atmicnetworks.com](#) Show history

'Unknown widget:\ 'volume sliderV*, data: (status: 'Error' , Reason: 'Unknown widget:\ 'volume sliderV**'

A. Widgets of type “Slider” are not supported on the device.

B. The UI extension was not exported to the device.

C. The name of the widget in the macro and the UI extension must match.

D. Promises are not supported for this device.

<https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/ce99/webex-board-administrator-guide-ce99.pdf>

Question: 30

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.

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

'Time span or seconds count' 'Guest Application Identifier'

'User Display Name' 'Guest User Key'

'Guest Application Secret' 'User Identifier'

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' }
);
```

'Time span or seconds count'

'Guest Application Identifier'

'User Display Name'

'Guest User Key'

'Guest Application Secret'

'User Identifier'

Reference:

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

Question: 31

DRAG DROP

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:

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.

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

Authentication credentials were missing or incorrect.

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

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: 32

DRAG DROP

Refer to the exhibit.

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

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".

```
curl -s -X [ ] "https://api.ciscospark.com/v1/[ ]" \
-H "Authorization: Bearer MDMYZTM1YTA_Pf84_1eb65fdf-ad72cae0e10f" \
-H "Content-Type: application/json" \
--data-raw '{"[ ]": "987654321", "[ ]": "[ ]", "isModerator": [ ]}'
```

personEmail

true

POST

memberships

roomId

Answer:

curl -s -X	POST	'https://api .ciscopark.com/v1 /		memberships		
-H "Authorization: Bearer MDMyZTM1YTA_Pf84_1eb65fdf-ad72cae0e10F\						
-H "Content-Type: application/json\"						
-data -raw 'f	roomId	": "987654321":	personEmail	": "alice@example.com": "isModerator":	true	

personEmail	true	POST	memberships	roomId
-------------	------	------	-------------	--------

Reference:

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

Question: 33

Refer to the exhibit.

```

"items":
{
  "id": "Y21zY29zcGFyazovL3VzLlLWT4Y2UtMTEzZjZjZmMxNGJl", "title":
  "DevNet Dev Support Questions", "lastActivity" : "2019-12-
13T21:32:11.424Z",
  "created": "2016-01-22T19:07:24.785Z",
b
{
  "id": "Y21zY29zcGFyazovL3VzWU1LWT0ZjQtZmJmMj13Y2ZmYWYz",
  "title": "DevNet Program Questions", "lastActivity" : "2019-12-
13T16:53:14.920Z", "created": "2016-02-24T18:52:35.013Z",
1
}
}

```

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

Answer: C

Reference:

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

Question: 34

DRAG DROP

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.

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

Answer:

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

Reference:

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

Question: 35

Refer to the exhibit.

```
const Webex = require('webex');

const webex = Webex.init({
  credentials: {
    access_token: 'A_VALID_ACCESS_TOKEN'
  }
});
```

When using the Webex Browser SDK to create calls and share screens, which two statements are valid given a 'webex' object such as displayed in the exhibit? (Choose two.)

- A. After a meeting is joined, it cannot be left programmatically until the host ends the meeting.
- B. The webex meetings.register() function must be invoked before attempting to join any meeting.
- C. The joinMeeting() function throws an error of type 'media stopped' if a media stream is stopped.
- D. Given a Webex meeting number the webex meetings.join() function can be used to join the meeting.
- E. The mediaSettings for a joined meeting accepts boolean attributes to send and receive audio, video, and screen share.

Answer: AB

Reference:

<https://developer.webex.com/docs/sdks/browser>

Question: 36

Refer to the exhibit.

:<https://api.ciscospark.com/v1/people?

max=1&cursor=bGltZXQ9MSZluZGV4PTI=>; rel="next" IContent-Type: application/json;charset=UTF-8

"notFoundIds":null, "items":[

"id": "Y21zY29zcGFyazovL3VzLlBFTllylmZjViLTQ40dltYjUyOS110WQ3NTRjTMjjiMzI". "emails": ["johnDoe@example.com"],

"displayName": "John Doe", "firstName": "John", "lastName": "Doe", }])

A Webex Teams REST API response is shown with the HTTP Header missing. Which HTTP header expected in this response?

- A. Push

- B. Link
- C. Patch
- D. Put

Answer: B

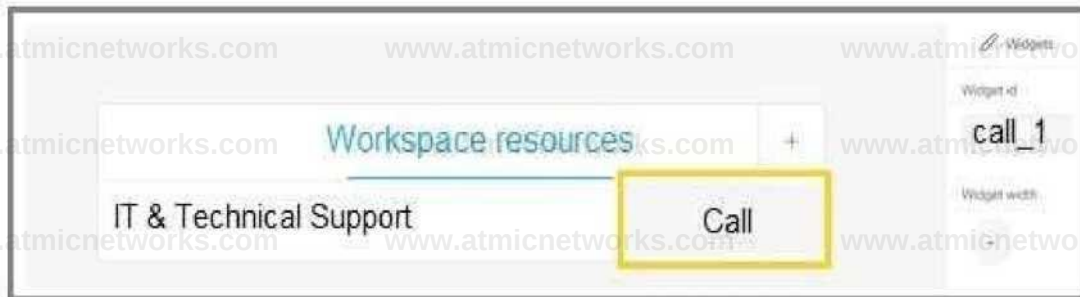
Reference:

<https://developer-portal-intb.ciscospark.com/docs/api/basics>

Question: 37

DRAG DROP

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

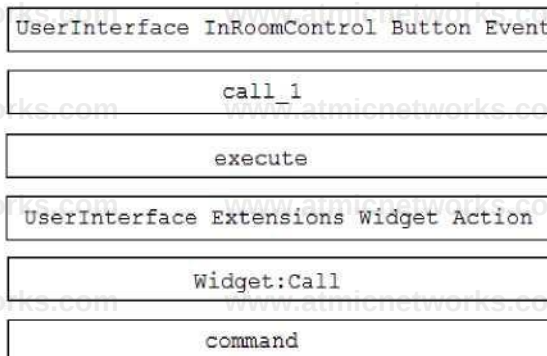
Answer:

```
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'));
});
```



Reference:

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

Question: 38

A company wants to adopt Webex Teams as a messaging platform and use REST APIs to automate the creation of teams and rooms. Which sequence of REST API requests is needed to create and populate a new Webex team and create a populated Webex room for the team?

- A. POST /teams, POST /memberships, POST /rooms
- B. POST /teams, POST /people, POST /rooms
- C. POST /teams, POST /team/memberships, POST /rooms
- D. POST /teams, POST /team/memberships, POST /rooms, POST /memberships

Answer: B

Reference:

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

Question: 39

Which Webex Teams webhook resource type indicates that a user interacted with a card?

- A. buttonActions
- B. attachmentActions
- C. webhookCardActions

Answer: B

Reference:

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

Question: 40

DRAG DROP

Drag and drop the code segments to construct a script that carries out these functions: adds a new room
adds two users, Bob, and Alice, to the room
when both the users have been added, sends a welcome message

```
const Webex = require('webex');
const webex = Webex.init({
  credentials: {
    access_token: "VALID TOKEN"
  }
});

[ ]

.then(room=>{

[ ]

[ ]

[ ]

})then( ()=> {

[ ]

});

});
```

```
webex.messages.create({
  markdown: '**Hi Everyone**',
  roomId: room.id
})
```

```
webex.membership.create({
  roomId: room.id,
  personEmail: 'bob@example.com'
})
```

```
return Promise.all([
```

```
webex.rooms.create(title: 'My First Room')])
```

```
webex.membership.create({
  roomId: room.id,
  personEmail: 'alice@example.com'
})
```

Answer:

```
const Webex = require('webex');
const webex = Webex.init({
  credentials: {
    access_token: "VALID TOKEN"
  }
});

webex.rooms.create({title: 'My First Room'})

.then(room=>{

  return Promise.all([

    webex.membership.create({
      roomId: room.id,
      personEmail: 'bob@example.com'
    })

    webex.membership.create({
      roomId: room.id,
      personEmail: 'alice@example.com'
    })

  ])then( ()=> {

    webex.messages.create({
      markdown: '**Hi Everyone**',
      roomId: room.id
    })

  });

});
```

```
webex.messages.create({
  markdown: '**Hi Everyone**',
  roomId: room.id
})
```

```
webex.membership.create({
  roomId: room.id,
  personEmail: 'bob@example.com'
})
```

```
return Promise.all([
```

```
webex.rooms.create({title: 'My First Room'})
```

```
webex.membership.create({
  roomId: room.id,
  personEmail: 'alice@example.com'
})
```

Reference:

https://github.com/webex/webex-js-sdk/tree/master/packages/node_modules/webex (see usage)

Question: 41

Refer to the exhibit.

```

1 ▼ /**
M * A small In-Room controls panel with 4 quick dial numbers
3 */
4 const xapi = require('xapi');
5
6 //These match the widget ids of the In-Room control buttons
7 v const numbers = {
8   dial_catering: 'it.',
9   dial_it_support: 'support',
10  dial_taxi: 'taxi example.com',
11  dial_reception: 'reception',
12 }, 13
14 function dial (number) {
15   xapi.command('dial', {number: number});
16 }
17
18 v function listen() {
19   xapi.event.on('Userinterface Extensions Widget Action', (event) => { if (event.Type === 'clicked') {
20     const number = numbers[event.WidgetId];
21     if(number) dial(number);
22     else console.log('Unknown button pressed', event.WidgetId);
23   }
24   }
25 },
26 );

```

Severity: _ Show history

22:38:35 [system] Starting macros...

22:38:35 Quick dial > Loading...

22:38:36 Quick dial Ready!

22:38:36 Quick dial 'Unhandled promise rejection* (code: 3, message: 'Ambiguous command*', data: {xpath: '/Call'}) }

An end user reports that the speed dial button is not working on their Webex Device, and when loading into the Macro Editor, this error was presented. On which line is the incorrect syntax?

- A. line 4
- B. line 14
- C. line 15
- D. line 22

Answer: C

Reference:

<https://community.cisco.com/t5/telepresence-and-video/ce9-2-1-macro-framework-discussions/td-p/3220093>

Question: 42

DRAG DROP

Drag and drop the code segments from the right of the screen into the targets on the left to create a script that retrieves a list of rooms using the Webex JavaScript API and outputs the title and ID to the console.

console for one that matches the title "My Webex discussion". Not all code segments are used.

```
var Webex = require("webex");
var webex = Webex.init({
  credentials: {accessToken: "jhgdhdiuytiut-hgjhgf"}
});

console.log(room.title.room.id);

»
catch(function(reason) { console.error(reason);
process.exit(1);
})
```

webex.rooms

```
.list({
  max: 500
})
```

```
var room = rooms.items.filter(function(room) {
  return room.title === "My Webex Discussion"
});
```

```
resolve(function(rooms) {
```

```
.getList({
  max: 500
})
```

webex.rooms.get

```
then(function(rooms) {
```

```
var room = rooms.items.filter(function(room) {
  return room.title === "MyWebex Discussion"
})[0];
```

Answer:

```

var Webex = require("webex")
var webex = Webex.init({
  credentials: {access_token: "jhgdhdiuytiut-hgjhg"}
});

webex.rooms

.list({
  max:500
})

.then(function(rooms) {

var room = rooms.items.filter(function(room) {
  return room title == "MyWebex Discussion"
})[0];

  console.log(room title.room.id)
})
.catch(function(reason) {
  console.error(reason);
  process.exit(1);
})

```

webex.rooms

```

.list({
  max:500
})

```

```

var room = rooms.items.filter(function(room){
  return title.room == "My Webex Discussion"
});

```

```

.resolve(function(rooms){

```

```

.getList({
  max:500
})

```

webex.rooms.get

```

.then(function(rooms) {

```

```

var room = rooms.items.filter(function(room) {
  return room title == "MyWebex Discussion"
})[0];

```

Reference:

<https://github.com/webex/webex-js-sdk/blob/master/README.md>

Question: 43

DRAG DROP

Refer to the exhibit.

```
{
  "messages": [
    {
      "$": {
        "xmlns:serv": "http://www.webex.com/schemas/2002/06/service",
        "xmlns:com": "http://www.webex.com/schemas/2002/06/common",
        "xmlns:ep": "http://www.webex.com/schemas/2002/06/service/ep",
        "xmlns:meet": "http://www.webex.com/schemas/2002/06/service/meeting",
        "header": {
          "response": {
            "result": "SUCCESS",
            "gsbStatus": "PRIMARY"
          }
        },
        "body": {
          "bodycontent": {
            "S": {
              "xs:type": "eplstRecordingResponse",
              "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
              "matchingRecords": {
                "total": "2",
                "returned": "2",
                "startFrom": "1"
              },
              "recording": [
                {
                  "recordingID": "90812441",
                  "hostWebExID": "jlev@cdpneighbors.com",
                  "name": "Jeff Levensailor's Personal Room-20190711 1154-1",
                  "createTime": "07/11/2019 08:06:27",
                  "timeZoneID": "11",
                  "size": "0.017041206",
                  "streamURL": "https://cdp.webex.com/wrldr.php?RCID=e7eba83fadfec6f5abd98637407499f7",
                  "fileURL": "https://cdp.webex.com/wrldr.php?RCID=dc37fb81c5ba41e42db43247ac95b7e8",
                  "recordingType": "5",
                  "duration": "9",
                  "format": "MP4",
                  "serviceType": "Meetingcenter",
                  "password": "4Mt3J3r4",
                  "passwordReq": "true",
                  "confID": "133167544940915779",
                  "shareToMe": "false",
                  "recordingID": "90812244",
                  "hostWebExID": "jlev@cdpneighbors.com",
                  "name": "Jeff Levensailor's Personal Room-20190711 1156-2",
                  "createTime": "07/11/2019 08:06:26",
                  "timeZoneID": "1T",
                  "size": "0.01679802",
                  "streamURL": "https://cdp.webex.com/wrldr.php?RCID=caf77cc9a8f564daeb06c6747c6eda0r",
                  "fileURL": "https://cdp.webex.com/wrldr.php?RCID=9a016ae489fe94af45b474c010047a8T",
                  "recordingType": "5",
                  "duration": "6",
                  "format": "MP4",
                  "serviceType": "Meetingcenter",
                  "password": "vHJpXCr4",
                  "passwordReq": "true",
                  "confID": "133167544940915779",
                  "shareToMe": "false"
                }
              ]
            }
          }
        }
      }
    }
  ]
}
```

A training coordinator must post links to Webex recordings on a company SharePoint site. This is usually a manual process, but a DevOps engineer wants to automate it using Webex XML APIs. After a successful LstRecording call wrapped in xml2js, the 'console.dir(result)' output is shown in the exhibit. Using 'dot notation', drag and drop the code below onto the code snippet to output the streamURL for each recording.

body	recording	bodyContent	message	streamURL
recordings = result ({		>>({		>>({
for (i=0, (recordings.length, i++)	[C ons ole. l og (re co rd i n g s[i]	{		)

Answer:

```
'recordings = result ({ body }){ { bodyContent } { streamURL } { message }
for (i=0, i<recordings.length, i++){Console.log(recordings[i] { { recording } } )}
```

body

recording

bodyContent

message

streamURL

Question: 44

Refer to the exhibit.

```
document.getElementById('share-screen').addEventListener('click', () => { if (activeMeeting) {
    const mediaSettings = { receiveShare: true, sendShare: true,

    console.info('SHARE-SCREEN: Preparing to share screen via 'getMediaStreams" ');
    activeMeeting.getMediaStreams(mediaSettings)
    // '[, localshare]' is grabbing index 1 from the mediaSettingsResultsArray // and storing it in a variable called localshare.
    .then((mediaSettingsResultsArray) => { const [, localshare] = mediaSettingsResultsArray;

    console.info('SHARE-SCREEN: Add local share via 'updateShare"');

    return « missing code » })
    .then(() => {
        console.info('SHARE-SCREEN: Screen successfully added to meeting. '); l)
    .catch(e) => (
        console.error('SHARE-SCREEN: Unable to share screen, error: '); console.error(e);

    } else { console.error('No active meeting available to share screen.');
```

Which code completes the return statement that initiates local screen sharing on the active meeting?

- A. activeMeeting.updateShare({ sendShare: true receiveShare: true, stream: null })
- B. activeMeeting.updateShare({ sendShare: true receiveShare: false, stream: remoteShare })
- C. activeMeeting.updateShare({ sendShare: true receiveShare: true, stream: localShare })
- D. activeMeeting.updateShare({ sendShare: false receiveShare: false, stream: null })

Answer: C

Reference:

https://github.com/webex/webex-js-sdk/tree/master/packages/node_modules/%40webex/plugin-meetings

Question: 45

Which list of API calls must be made to provision 50 new Webex Room devices for an organization?

- A. POST to /places with 'displayName', POST the returned 'placeld' to /devices/
- B. POST 'displayName' to /places, POST the returned 'placeld' to /devices/activationCode
- C. POST to /licenses, POST the returned 'liceneseld' and a specified 'displayName' to /devices
- D. POST to /licenses, POST 'displayName' to /places, POST 'licensed' and placeld' to /devices

Answer: A

Reference:

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

Question: 46

Which two tasks should use a Webex bot instead of aWebex integration? (Choose two.)

- A. Ask a Question: in natural language to obtain the current price of a stock.
- B. Translate a word or phrase from Italian to English.
- C. Notify all users who send a 1:1 message that the user is currently out of the office.
- D. Automatically delete a message that was sent with spelling or grammatical errors.
- E. Archive all the messages in a group space.

Answer: AB

Reference:

<https://developer.webex.com/docs/bots>

Question: 47

DRAG DROP

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

```
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');
```

xCommand

command

jsxapi

devices_read

printHttp

catch

Answer:

```

const xapi = require (' cormrand ') J
function printHttp(url) {
  xapi. comir.and ('HTTPClient Get'. {Uri: url })
    .then((data) => { console.Log({ data:JSON.parse.body(data) , });
    })
    . catch ( () -> {
      console.error({
        error: 'HTTP Get failed'.
      })
    })
}

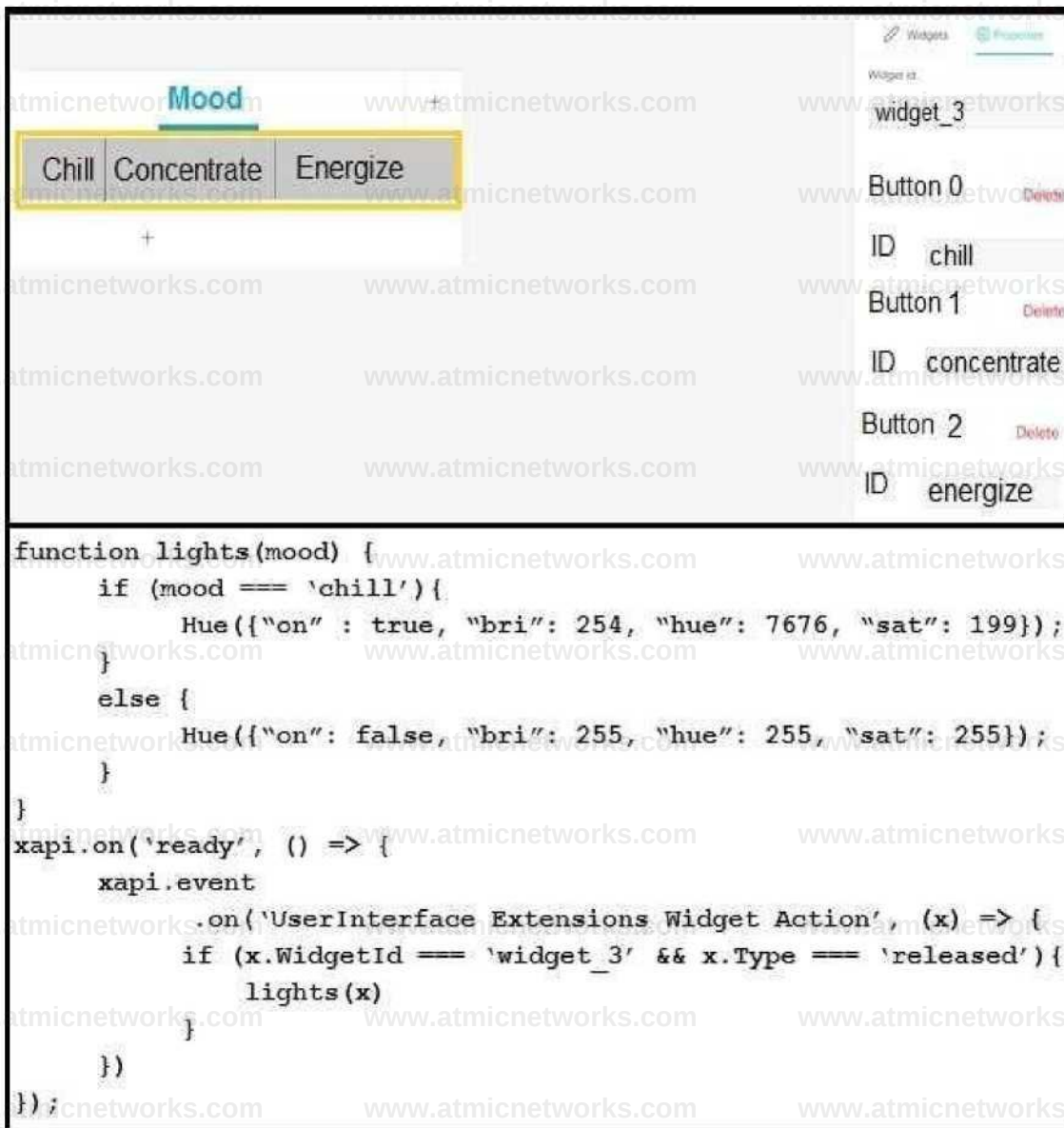
printHttp ('http://example.com')

```

xCommand	command
jsxapi	devices_read
printHttp	catch

Question: 48

Refer to the exhibit.



Certain lighting conditions are needed when hosting Webex meetings for a particular department in a company. A split button that integrates with the lighting controls is added to the Touch 10. However, when the user selects Chill, the lights turn off completely. Which code change resolves this issue?

- A. Change xapi.event to xapi.httpfeedback.
- B. Set x.Type to 'pressed' instead of 'released'.
- C. Set x.Type to 'clicked' instead of 'released'.
- D. Pass x.Value instead of x to the lights function.

Answer: B

Reference:

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

Question: 49

Which two statements are true when comparing the SSH, HTTP, and WebSocket interfaces of a Cisco Collaboration device? (Choose two.)

- A. The SSH interface supports XML payloads only.
- B. The SSH, HTTP, and WebSocket interfaces support JSON and XML payloads.
- C. The HTTP interface supports JSON and XML payloads.
- D. Communications can be encrypted for all SSH, HTTP, and WebSocket interfaces.
- E. The WebSocket interface supports JSON payloads only.

Answer: BD

Question: 50

Which two statements about incoming WebEx call notifications on iOS and Android applications are true? (Choose two.)

- A. When running in the background on Android, Firebase Cloud Messaging must use a notification message.
- B. When running in the background on Android, Firebase Cloud Messaging must use a data message.
- C. Webhooks are required with the iOS SDK only when the application is in the background.
- D. When running in the background on iOS, Application Push Notifications must be configured to be silent.
- E. Webhooks are required in the foreground and background for iOS and Android applications.

Answer: BC

Reference:

<https://github.com/webex/webex-android-sdk-example-push-notification-server/blob/master/README.md> <https://developer.webex.com/docs/sdks/ios>

Question: 51

DRAG DROP

Drag and drop the code to complete the snippet that sends a message to the room "My First Room" using the Browser SDK. Not all options are used.

```
var webex = require ('webex/env');
webex.rooms.list ({
    max: 10
})
    .then(function(rooms) {
        var room = rooms.items.filter(function (room) { return room.title ===
            'My First Room!';
```

:o:?

```
return webex.  
  
    'Hello World!  
  
    room.id  
  
    })  
  
    //Make sure to log errors in case something goes wrong  
    .catch (function(reason) {  
        console.error(reason);  
        process.exit(1);  
    })
```

send	text:
create	roomId:
messages	message:

Answer:

```
var webex = require ('webex/env');  
webex.rooms.list({ max: 10  
    . then, (function (rooms) {  
        var room = rooms.items.filter(function (room) { return  
            room.title === 'My First Room!';  
        }) [0];
```

```
return webex. messages . create
```

```
    text: 'Hello World!',  
    roomId: room.id
```

```
    });  
    })  
  
    //Make sure to log errors in case something goes wrong.  
    .catch (function(reason) {  
        console.error(reason);  
        process.exit(1);  
    });
```

send	text:
create	roomId:
messages	message:

Reference:

<https://developer.webex.com/docs/sdks/node>

Question: 52

Which two statements about using the Compliance Office role to manage data in Webex Teams are true? (Choose two.)

- A. For group rooms with members from multiple organizations, only the compliance officer of the owning organization can manage all data.
- B. The compliance officer can manage data only for users who have granted the spark-compliance messages scope.
- C. For direct rooms between two organizations, each compliance officer can manage data only from their organization.
- D. For group rooms with members from multiple organizations, either compliance officer can manage all data.
- E. For direct rooms between two organizations, the compliance officer of either organization can manage all data.

Answer: CE

Reference:

<https://developer-portal-intb.ciscospark.com/docs/api/guides/compliance>

Question: 53

Refer to the exhibit.

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

```
// Start a call
```

```
*
```

```
*
```

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

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'})
```

```
xapi.command('SIP', {Number: 'jdoe@example.com'})
```

```
xapi.command('Dial', {Number: 'jdoe@example.com'})
```

A. OptionA B. OptionB C. OptionC D. Option D

Answer: D

Reference:

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

Question: 54

DRAG DROP

Drag and drop the expressions to create a Webex Teams widget that uses Guest Issuer to enable customers to chat and meet with agents. Not all options are used.

```
<script>
$(document).ready(function () {
  $("#btn").click(function () {
    var widgetEL = document.getElementById('supportChart')
    $.ajax({
      url: '/api/getSession', type: 'POST', contentType: 'application/json', data: JSON.stringify({ name: $("#name").val(),
      email: $("#email").val() H }).done(function (session) {
        ciscospark.widget(widgetEL).spaceWidget({ guestToken: |_____|
        destinationId: |_____|
        destinationType: |_____|
        activities: { 'files': false, "meet": true, "message": true, "people":
        false),
        InitialActivity: |_____|
        secondaryActivitiesFullWidth: false });
        | "userid"
        | "message"
        session.token
        Direct
        |session.agent|
        guest
      })
    })
  })
  app.post('/api/getSession', function(req, res) { res.status(200).send({
    agent: 'johndoe@example.com',
    token: jwt.sign({
      "sub": '{req.body.email}', "name": '{req.body.name}', "iss": '{issuer}'
    }, Buffer.from('{secret}', 'base64'), {})
```

Answer:

```

<script>
$(document).ready(function () {
  $("#btn").click(function () {
    var widgetEL = document.getElementById( 'supportchart');
    $.ajax({
      url: 7api/getSession'.
      type: 'POST',
      contentType: 'application/json',
      data: JSON.stringify({
        name: $("#name").val(),
        email: $("#email").val()
      })
    }).done(function (session) {
      ciscospark.widget(widgetEL).spaceWidget({
        guestToken:
        destinationId:
        destination Type: activities: {"files":
        initialActivity:
        secondaryActivitiesFullWidth: fals });
    })
  })
})
</script>
app.post(7api/getSession', function(req, res) { res.status(200).send({
  agent: 'johndoe@example.com',
  token: jwt.sign({
    "sub": '${req.body.email}',
    "name": ${req.body.name}', "iss": '{{issuer}'
  }, Buffer.from('${secret}', 'base64}, {}
  });
});
});

```

"userid"
 "message"
 session.token
 Direct
 session.agent
 guest

Question: 55

Which user role implicitly has permission in the EventsAPI 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

Answer: A

Reference:

Question: 56

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

- A. <link rel= "stylesheet" href= "<https://code.s4d.io/widget-recents/production/main.css>">
- B. <script src= "stylesheet" href= "<https://code.s4d.io/widget-space/production/main.css>">
- C. <link rel= "stylesheet" href= "<https://code.s4d.io/widget-recents/production/bundle.js>">
- D. <link rel= "stylesheet" href= "<https://code.s4d.io/widget-space/production/main.css>">

Answer: D

Reference:

<https://developer.webex.com/docs/widgets>

Question: 57

Refer to the exhibit.

```

1  const request = require('request-promise').defaults({
2      json: true,
3      baseUrl: 'https://api.ciscospark.com/v1/'
4  }
5
6  const dontsay = [ 'swag', 'fleek', 'stay', 'shade', 'bae', 'yas', 'lit', 'ratchet' ]
7
8  function isSensitive(text) {
9      sensitive = false
10     for (j = 0 ; j < dontsay.length ; j++) {
11         if (text Si text).includes(dontsay[j])H (
12             sensitive = true
13         }
14     )
15     return sensitive
16 }
17
18 function getMessages() {
19     request.get({
20         url: *.....":
21         headers: (Authorization: 'Bearer S(compliance token|J
22     }).then(events => {
23         for (j ** 0; i < events.items.length; i++) {
24             if (events.items[i].data.text && isSensitive(events.items[i].data.text)) (
25                 request.post((
26                     uri: 'messages',
27                     headers: (Authorization: 'Bearer $)bot token)'),
28                     body: {
29                         'toPersonId': '$(events.items[i].actorId)',
30                         'markdown': 'Please don't say *$$(events.items[i].data.text)** ever again..'
31                     },
32                 ))
33             }
34         }
35     })
36 }
37
38 getMessages()
39
40

```

A company uses Webex Teams extensively for communications involving customers, and want to enforce a

consistent messaging policy. Which code completes line 20 to send a notification when noncompliant messages are detected?

- A. `events?resource=people`
- B. `compliance?resource=messages`
- C. `events?resource=messages`
- D. `compliance?resource=people`

Answer: C

Explanation:

The function is `getmessages`. Then a request is made so the `get` comes in action and does `events?`

`Resource=messages` to get the data for the events.

Question: 58

Which XML snippet does an admin use with a Webex Meetings XML API request to retrieve info on multiple users?

- A. `<bodyContent xsi:type="java:com.webex.service.binding.user.GetUser">
<webExId>.*</webExId>
</bodyContent>`
- B. `<bodyContent xsi:type="java:com.webex.service.binding.user.LstsummaryUser">
<webExId>.*</webExId>
</bodyContent>`
- C. `<bodyContent xsi:type="java:com.webex.service.binding.user.GetUser">
</bodyContent>`
- D. `<bodyContent xsi:type="java:com.webex.service.binding.user.LstsummaryUser">
</bodyContent>`

Answer: A

Reference:

<https://developer.cisco.com/docs/webex-xml-api-reference-guide/#!getuser>

Question: 59

DRAG DROP

The Express framework receives a Webex Teams webhook event when someone presses a button in an Adaptive Card. Drag and drop the property names onto the code snippet to access the form field named "preference". Not all options are used.

```

app.use(express.json());
app. [ ] ('/cardcatcher', function(req, res){
  webex.attachmentActions. [ ] (req.body. [ ] .id)
  .then(cardSubmission => {
    lunchPreference = cardSubmission. [ ] .preference;
  })
});

```

Answer:

```

app.use(express.json());
app. [ post ] ('/cardcatcher', function(req, res){
  webex.attachmentActions. [ action ] (req.body. [ form ] .id)
  .then(cardSubmission => {
    lunchPreference = cardSubmission. [ data ] .preference;
  })
});

```

Reference:

<https://github.com/marchfederico/ciscospark-websocket-events>

Question: 60

Which two statements describe valid limitations for Webex Teams bot accounts in a group space? (Choose two.)

- A. A bot account cannot send a message without an SSL certificate.
- B. A bot account cannot hear a message unless the bot is @ mentioned.
- C. A bot account cannot send a message without a valid access token.
- D. A bot account cannot respond to another bot.
- E. A bot account cannot send a message with an @mention.

Answer: BC

<https://developer.webex.com/docs/bots>