



300-920^{Q&As}

Developing Applications for Cisco Webex and Webex Devices
(DEVWBX)

Pass Cisco 300-920 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

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

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cisco
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

```
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 && text.includes(dontsay[j])) {
12      sensitive = true
13    }
14  }
15  return sensitive
16 }
17
18 function getMessages() {
19   request.get({
20     url: ".....",
21     headers: {Authorization: 'Bearer ${compliance_token}' }
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 getMessages()
38
39
40
```

Refer to the exhibit. 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

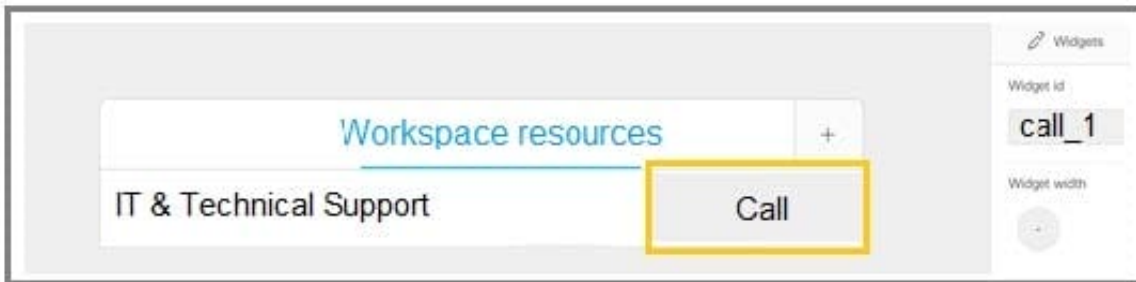
Correct Answer: C



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 2

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.

Select and Place:

```
const xapi = require('xapi');
xapi.event.on(' [ ] ', (event)=> {
  if (event.WidgetId !== ' [ ] ') return;
  if (event.Type !== 'clicked') return;
  xapi. [ ]
  (' [ ] ', (Number: '1000'));
});
```

- UI InRoomControl Button Event
- call_1
- execute
- UI Extensions Widget Action
- Widget:Call
- command

Correct Answer:



```
const xapi = require('xapi');  
xapi.event.on('  ', (event)=> {  
  if (event.WidgetId !== '  ') return;  
  if (event.Type !== 'clicked') return;  
  xapi.   
  ('  ', (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 3



The screenshot shows the 'UI Extensions configuration - Edited' interface. At the top, there's a 'Settings' button and a '+' icon. Below that, a 'Volume' label is next to a slider widget, which is highlighted with a yellow box. To the right, a 'Properties' panel shows 'Widget id' set to 'volume_slider'. Below the slider, there are '-' and '+' buttons and a '4' in the center. The bottom half of the image shows a code editor with the following JavaScript code:

```
1  const xapi = require('xapi');
2
3  xapi.on('ready', () => {
4    console.log('macro started');
5
6    xapi.status.get("Auto Volume").then((volume) => {
7      xapi.command('UserInterface Extensions widget SetValue', {
8        WidgetId: 'volume_slider',
9        Value: volume* 255 / 100
10     });
11   });
12
13  xapi.event.on('UserInterface Extensions Widget Action', (event) => {
14    if ((event.WidgetId === 'volume_slider') && (event.Type === 'released')) {
15      const volume = Math.round(parseInt(event.Value) * 100/255);
16      xapi.command('Auto Volume Set', {Level: volume });
17      return;
18    }
19  });
20 });
```

Below the code editor, there's a console output showing an error:

```
18:42:36 exam > 'macro started'
18:42:36 exam > 'Unhandled precise rejection' (code: 0,
  message: 'Unknown widget:\ 'volume_slider\'',
  data: {status: 'Error', Reason: 'Unknown widget:\ 'volume_slider\''})
```

Refer to the exhibit. A macro and a UI extension (also called In-Room Control) are being developed. What is the reason for the error displayed in the console?

- 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.

Correct Answer: B

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

QUESTION 4



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.

Select and Place:

```

<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()
      })
    })
  }).done(function (session) {
    ciscospark.widget(widgetEl).spaceWidget({
      guestToken: ,
      destinationId: ,
      destinationType: ,
      activities: {"files": false, "meet": true, "message": true, "people": false},
      initialActivity: ,
      secondaryActivitiesFullWidth: false
    });
  })
})
</script>

```

```

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'), {})
  });
});

```

"userId"

"message"

session.token

Direct

session.agent

guest

Correct Answer:



```
<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()
      })
    })
  }).done(function (session) {
    ciscopark.widget(widgetEL).spaceWidget({
      guestToken: 
      destinationId: 
      destinationType: 
      activities: {"files": false, "meet": true, "message": true, "people": false},
      initialActivity: 
      secondaryActivitiesFullWidth: false
    });
  })
})
</script>
```

```
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'), {})
  });
})
```

QUESTION 5



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

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

Refer to the exhibit. 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.

Correct Answer: AB

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

[300-920 VCE Dumps](#)

[300-920 Exam Questions](#)

[300-920 Braindumps](#)