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

QUESTION 1

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'});
});
```

UserInterface InRoomControl Button Event

call_1

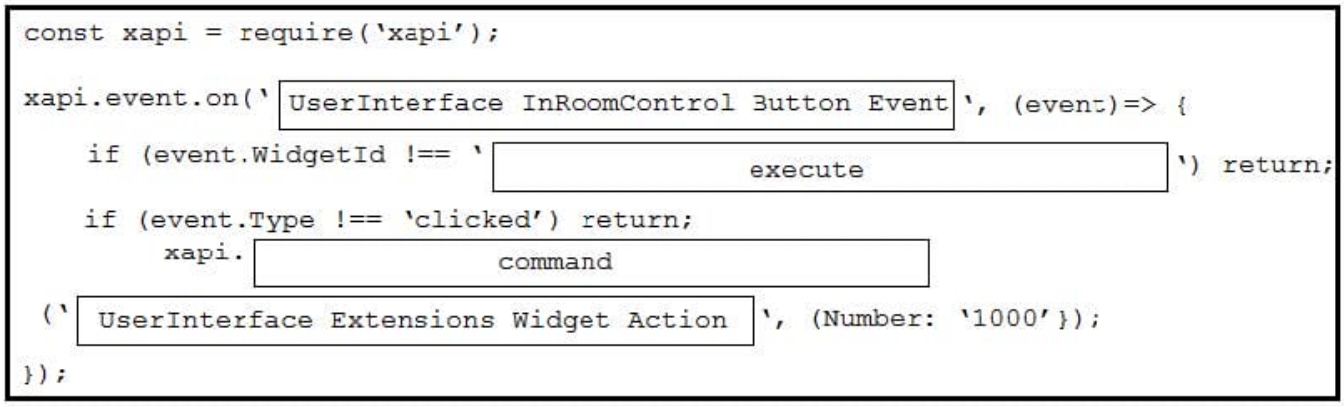
execute

UserInterface Extensions Widget Action

Widget:Call

command

Correct Answer:



call_1

Widget:Call

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 2

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.

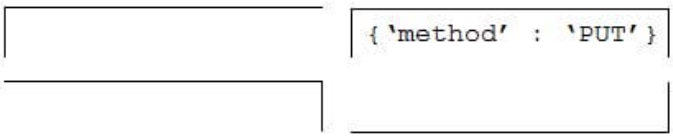
Select and Place:

```
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>
             <                          >
 PD4WiqNfRkxBR19B...RERJTkdfU0hBMjU2X0FMR09SSVRITV8=
 </                          >
       </securityContext></header>
       <body>
 <bodyContent xsi:type= "java.com.webex.service.binding.meeting.CreateMeeting">
           <metaData><confName>Sample Meeting</confName>
           <meetingType>105</meetingType></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',
                          );
 req.write(xml);
 req.end();
```

{'method' : 'POST'}	{'method' : 'PUT'}
accessToken	sessionTicket

Correct Answer:


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>
< {'method' : 'POST'} >
PD4WiqNfRkxBR19B...RERJTkdfU0hBMjU2X0FMR09SSVRITV8=
</ accessToken >
</securityContext></header>
<body>
<bodyContent xsi:type= "java.com.webex.service.binding.meeting.CreateMeeting">
<metaData><confName>Sample Meeting</confName>
<meetingType>105</meetingType></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();


{'method' : 'PUT'}

QUESTION 3

```
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

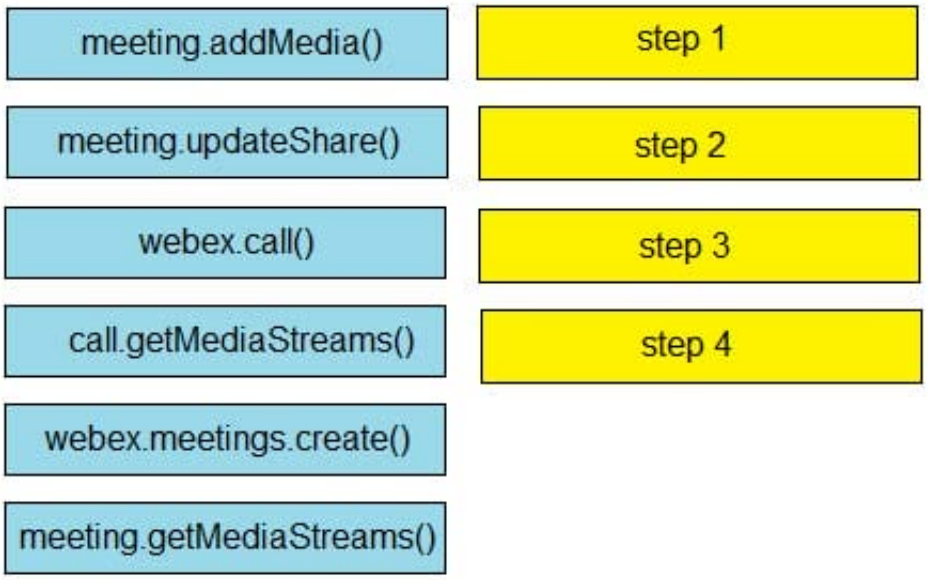
Correct Answer: B

QUESTION 4

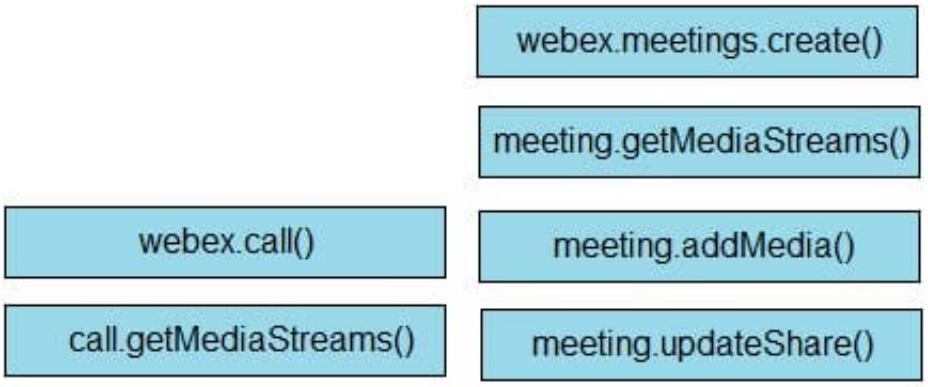
DRAG DROP

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.

Select and Place:



Correct Answer:



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

QUESTION 5

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:

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/WBXService/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.getElementByTagNameNS(
                 `  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