QUESTION 53
Drag and Drop Question
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">
    <header><securityContext>
        <siteName>apidemoeu</siteName>
        <webExID>alice</webExID>
        <sessionTicket>AAABb6CkTRgAABUYAR2AD2X0FMR09SSVRITV8=</sessionTicket>
    </securityContext></header>
    <body>
        <bodyContent xsi:type="        ⬚        ">

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

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>AAABb6CkTRgAABUYAR2AD2X0FMR09SSVRITV8=</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        ',init)
           Service/XMLService
    .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/meeting
    'fileURL') [0].textContent});
```

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

```
http://www.webex.com/schemas/
2002/06/service/ep
```

```
method: 'GET'
```

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

QUESTION 54
Drag and Drop Question
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.

**Braindump2go Guarantee All Exams 100% Pass
One Time!**

```
const jwt = require('jsonwebtoken)
let token = jwt.sign(
    {
        'sub': [                    ] ,

        'name': [                    ] ,

        'iss': [                    ] ,

    },
    Buffer.from([                    ] ,'base64'),

    { expressIn: [                    ] }

);
```

'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'),

    { expressIn: [ 'Time span or seconds count' ] }

);
```

'Guest User Key'

Explanation:
https://developer.webex.com/docs/guest-issuer

300-920 Exam Dumps 300-920 Exam Questions 300-920 PDF Dumps 300-920 VCE Dumps

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

QUESTION 55
Drag and Drop Question
Drag and drop the definitions from the left onto the correct Webex Teams REST API HTTP response status codes on the right.

Definition	Code
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:

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.

Explanation:
https://developer.webex.com/docs/api/basics

QUESTION 56
Drag and Drop Question
Refer to the exhibit. 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".

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

```
curl –s –X [        ]  "https://api.ciscospark.com/v1/ [        ]  "\
-H "Authorization: Bearer MDMyZTM1YTA_PF84_1eb65fdf-ad72cae0e10f"\
-H "Content-Type: application/json"\
-data-raw '(" [        ]  ". "987654321"." [        ]  "."alice@example.com"."isModerator"." [        ] ")'
```

| personEmail | true | POST | memberships | roomId |

Answer:

```
curl –s –X [ POST ]  "https://api.ciscospark.com/v1/ [ memberships ]  "\
-H "Authorization: Bearer MDMyZTM1YTA_PF84_1eb65fdf-ad72cae0e10f"\
-H "Content-Type: application/json"\
-data-raw '(" [ roomId ]  ". "987654321"." [ personEmail ]  "."alice@example.com"."isModerator"." [ true ] ")'
```

Explanation:

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

QUESTION 57
Drag and Drop Question
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:

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

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

QUESTION 58
Drag and Drop Question
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'});
});
```

call_1

Widget:Call

Explanation:

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

QUESTION 59
Drag and Drop Question
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
})
```

```
});
```

```
});
```

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

QUESTION 60
Drag and Drop Question
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 for one that matches the title "My Webex discussion". Not all code segments are used.

```
webex.rooms
```

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

```
.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
```

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

```
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-hgjhgf'}
});
```

```
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);
})
```

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

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

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

```
webex.rooms.get
```

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

QUESTION 61
Drag and Drop Question
Refer to the exhibit. 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 sucessful 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.

```
{messages
 { "$":
   {'xmlns:serv': 'http://www.webex.com/schemas/2002/06/service',
    'xmlns:com': 'http://www.webex.com/schemas/2002/06/common',
    'xmlns:ep': 'http://www.webex.com/schemas/2002/06/service/ep',
    'xmlns:meet': 'http://www.webex.com/schemas/2002/06/service/meeting' },
   header: { response: { result: 'SUCCESS', gsbStatus: 'PRIMARY' } },
   body:
   { bodyContent:
     { '$':
         { 'xsi: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 Levesailor's Personal Room-20190711 1154-1",
          createTime: '07/11/2019 08:06:27',
          timeZoneID: '11',
          size: '0.017041206',
          streamURL: 'https://cdp.webex.com/wr/ldr.php?RCID=e7eba83fadfec6f5abd98637407499f7",
          fileURL: 'https://cdp.webex.com/wr/lsr.php?RCID=dc37fb81c5ba41e42db43247ac95b7e8',
          recodringType: '5',
          duration : '9',
          format : 'MP4'
          serviceType : 'MeetingCenter',
          password : '4Mt3J3r4',
          passwordReq : 'true',
          confID : '133167544940915779',
          shareToMe : 'false' },
          recordingID : '90812244',
          hostWebExID : 'jlev@cdpmeighbors.com'
          name : "Jeff Levensailor's Personal Room-20190711 1156-2",
          createTime: '07/11/2019 08:06:26',
          timeZoneID: '11',
          size: '0.01679802',
          streamURL: 'https://cdp.webex.com/wr/ldr.php?RCID=caf77cc9a8f564daeb06c6747c6eda01',
          fileURL: 'https://cdp.webex.com/wr/lsr.php?RCID=9a016ae489fe94af45b474c010047a81',
          recordingType : '5',
          duration : '6',
          format : 'MP4',
          serviceType : 'MeetingCenter',
          password : 'vHJpXCr4',
          passwordReq : 'true',
          confID : '133167544940915779',
          shareToMe : 'false' }]}}}}
```

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

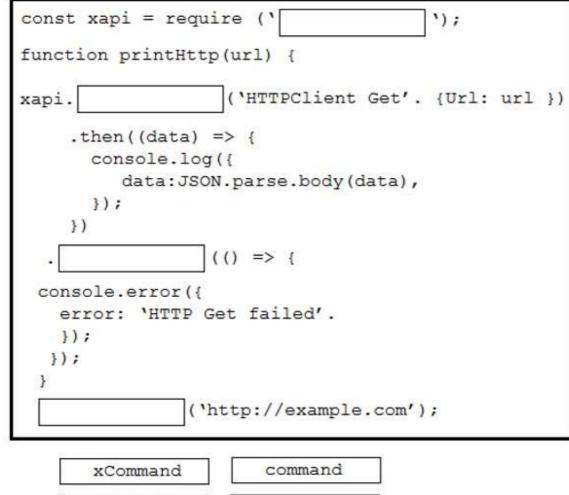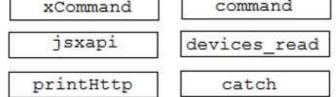| body | recording | bodyContent | message | streamURL |

Answer:

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

QUESTION 62
Drag and Drop Question
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 ('[ command ]');

function printHttp(url) {

xapi.[ command ]('HTTPClient Get'. {Url: url })

    .then((data) => {
      console.log({
          data:JSON.parse.body(data),
      });
    })
  .[ catch ](() => {
  console.error({
    error: 'HTTP Get failed'.
    });
  });
}
  [ printHttp ]('http://example.com');
```

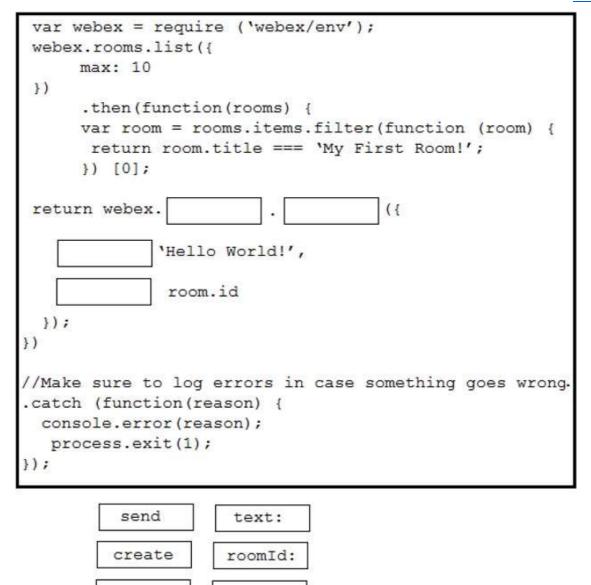|	
xCommand	command
jsxapi	devices_read
printHttp	catch

QUESTION 63
Drag and Drop Question
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!';
    }) [0];

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

message:

Explanation:
https://developer.webex.com/docs/sdks/node