- **Vendor:** Cisco
- **Exam Code:** 300-920
- **Exam Name:** Developing Applications for Cisco Webex and Webex Devices (DEVWBX)
- **New Updated Questions from Braindump2go (Updated in August/2020)**

QUESTION 39
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
Explanation:
https://developer.webex.com/blog/real-world-walkthrough-of-building-an-oauth-webex-integration

QUESTION 40
Where does the WebEx node reside?

A. only in the DMZ
B. on the Enterprise Branch
C. anywhere on the network
D. near the Meeting Director

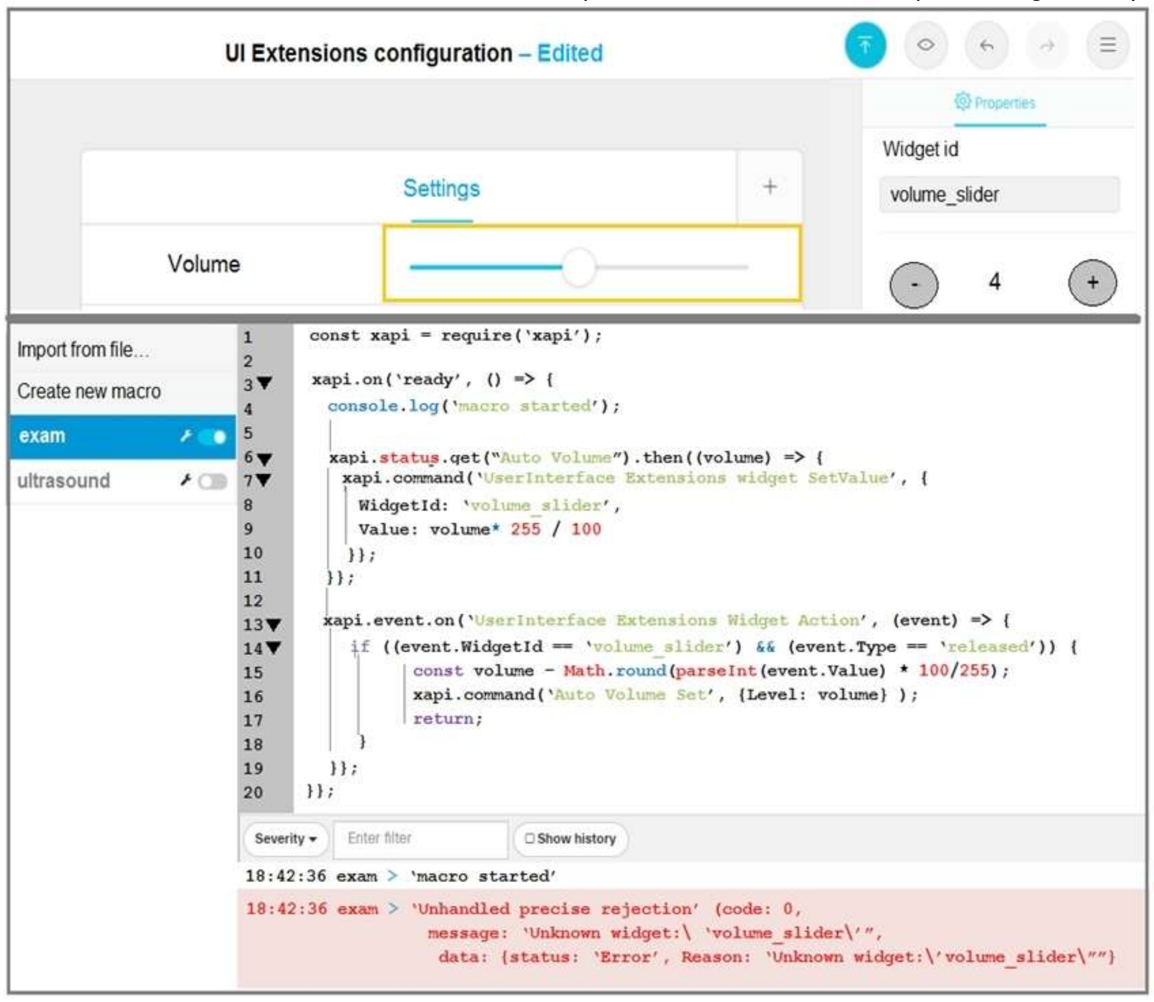
Answer: C

QUESTION 41
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
Explanation:
https://developer.webex.com/docs/api/guides/webhooks

QUESTION 42
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?



```
const xapi = require('xapi');

xapi.on('ready', () => {
  console.log('macro started');

   xapi.status.get("Auto Volume").then((volume) => {
    xapi.command('UserInterface Extensions widget SetValue', {
      WidgetId: 'volume_slider',
      Value: volume* 255 / 100
    });
  });

 xapi.event.on('UserInterface Extensions Widget Action', (event) => {
    if ((event.WidgetId == 'volume_slider') && (event.Type == 'released')) {
          const volume - Math.round(parseInt(event.Value) * 100/255);
          xapi.command('Auto Volume Set', {Level: volume} );
          return;
     }
   });
});
```

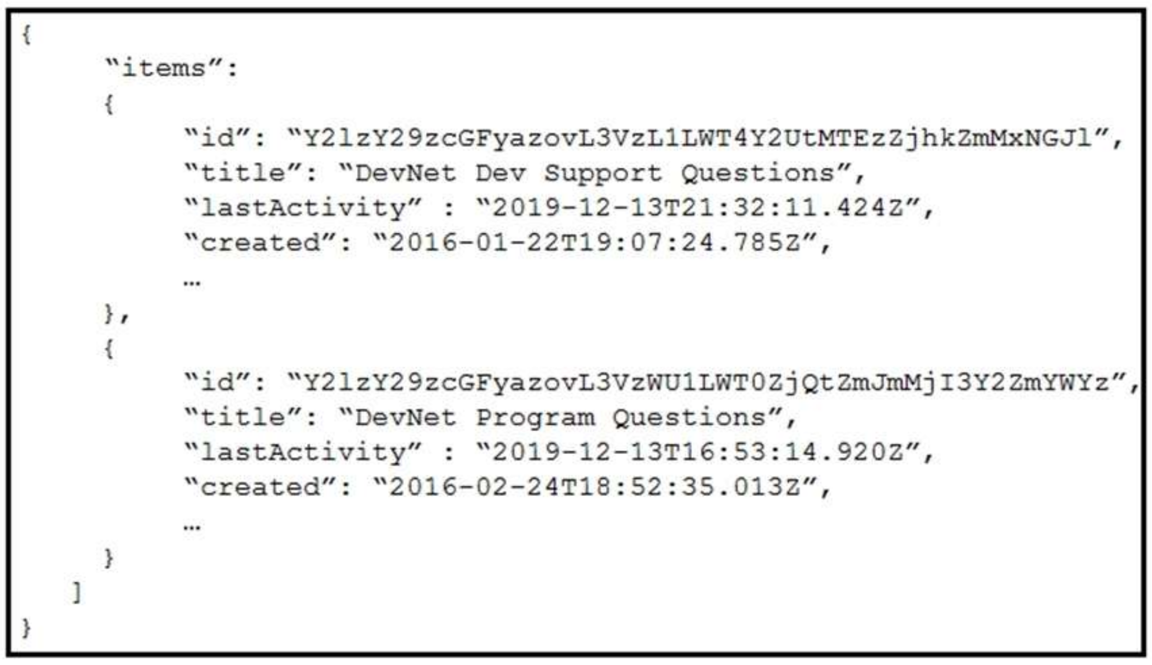
```
18:42:36 exam > 'macro started'
18:42:36 exam > 'Unhandled precise rejection' (code: 0,
                  message: 'Unknown widget:\ 'volume_slider\'",
                   data: {status: 'Error', Reason: 'Unknown widget:\'volume_slider\""}
```

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.

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

QUESTION 43
Refer to the exhibit. Which Webex Teams REST API request has generated the response body in the exhibit?

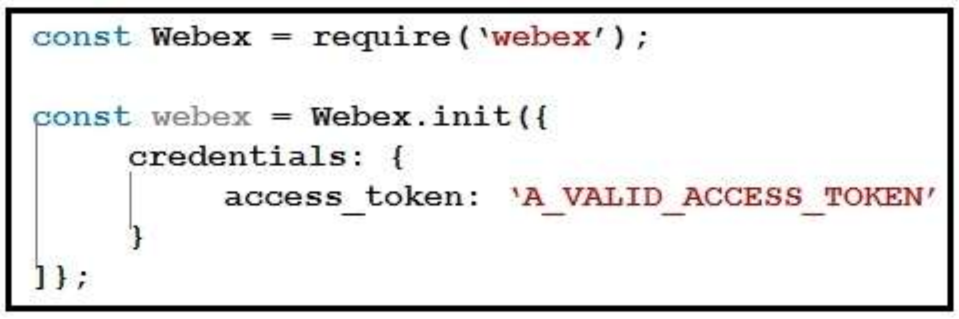


```
{
     "items":
     {
          "id": "Y2lzY29zcGFyazovL3VzL1LWT4Y2UtMTEzZjhkZmMxNGJl",
          "title": "DevNet Dev Support Questions",
          "lastActivity" : "2019-12-13T21:32:11.424Z",
          "created": "2016-01-22T19:07:24.785Z",
          …
     },
     {
          "id": "Y2lzY29zcGFyazovL3VzWU1LWT0ZjQtZmJmMjI3Y2ZmYWYz",
          "title": "DevNet Program Questions",
          "lastActivity" : "2019-12-13T16:53:14.920Z",
          "created": "2016-02-24T18:52:35.013Z",
          …
     }
   ]
}
```

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

QUESTION 44
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.)

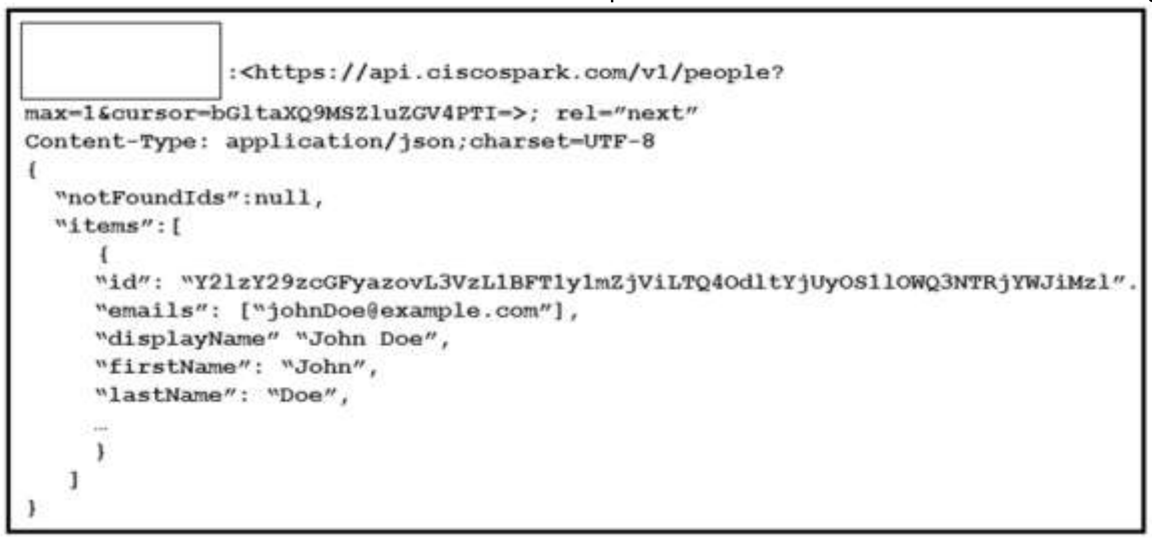


```
const Webex = require('webex');

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

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

QUESTION 45
Refer to the exhibit. A Webex Teams REST API response is shown with the HTTP Header missing. Which HTTP header expected in this response?



```
          :<https://api.ciscospark.com/v1/people?
max=1&cursor=bGltaXQ9MSZluZGV4PTI=>; rel="next"
Content-Type: application/json;charset=UTF-8
{
  "notFoundIds":null,
  "items":[
     {
     "id": "Y2lzY29zcGFyazovL3VzL1BFTlylmZjViLTQ4OdltYjUyOS1lOWQ3NTRjYWJiMzl".
     "emails": ["johnDoe@example.com"],
     "displayName" "John Doe",
     "firstName": "John",
     "lastName": "Doe",
     …
     }
   ]
}
```

A. Push
B. Link
C. Patch
D. Put

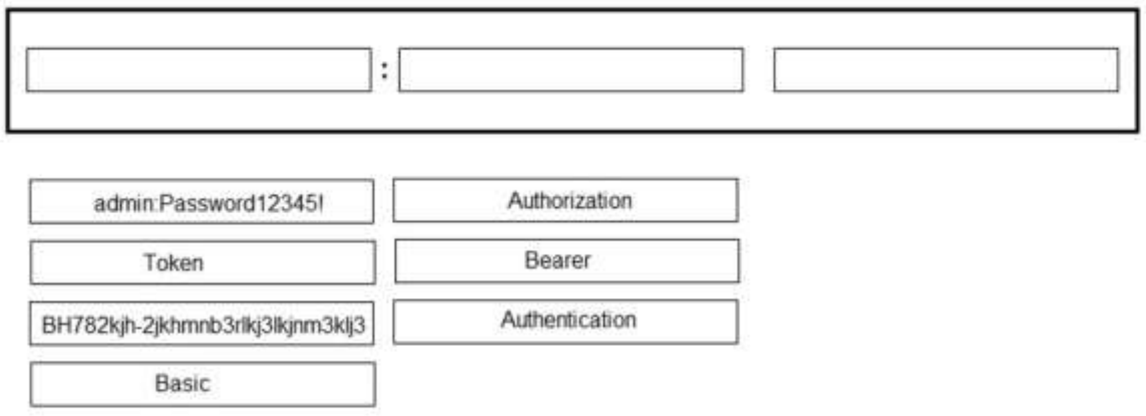
Answer: B
Explanation:
https://developer-portal-intb.ciscospark.com/docs/api/basics

QUESTION 46
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?

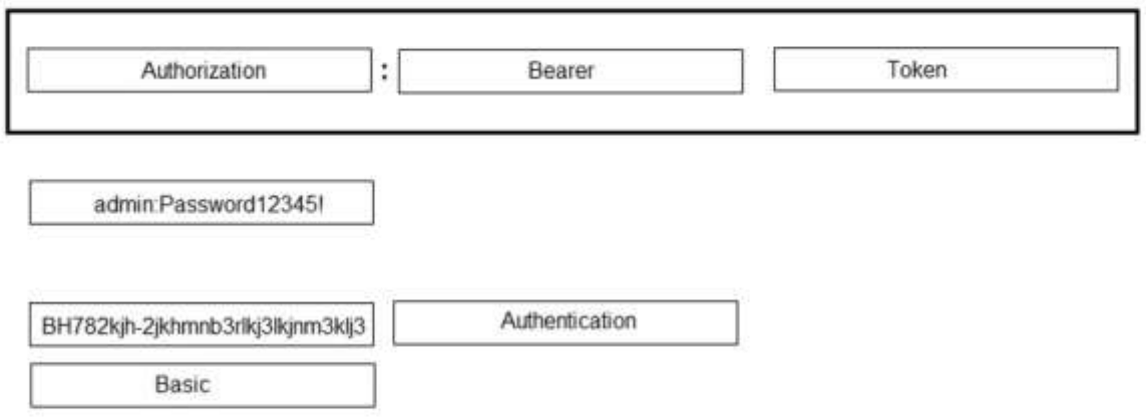
A. POST /teams, POST /memberships, POST /rooms
B. POST /teams, POST /people, POST /rooms
C. POST /teams, POST /team/memberships, POST /rooms
D. POST /teams, POST /team/memberships, POST /rooms, POST /memberships

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

QUESTION 47
Drag and Drop Question
Drag and drop the components to create the authentication header used for the Webex Teams API. Not all options are used.

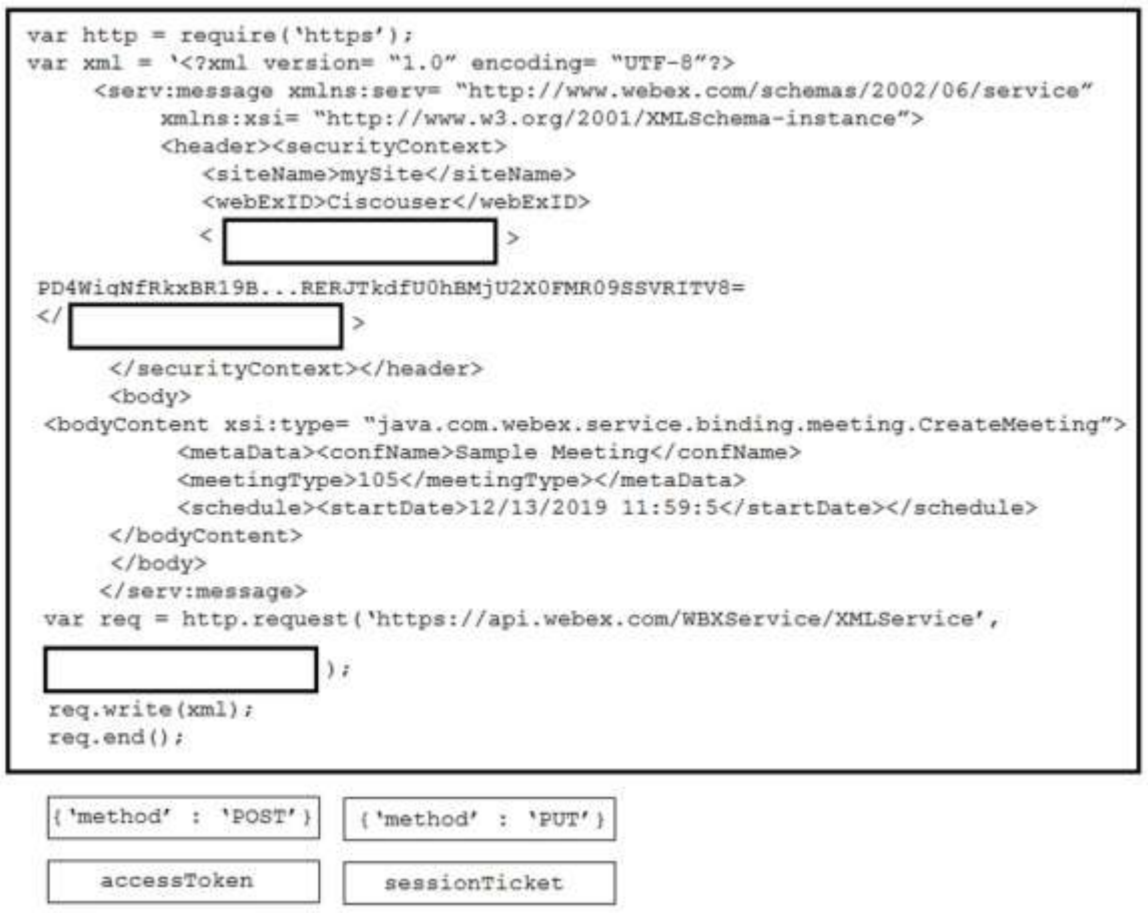


Answer:



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

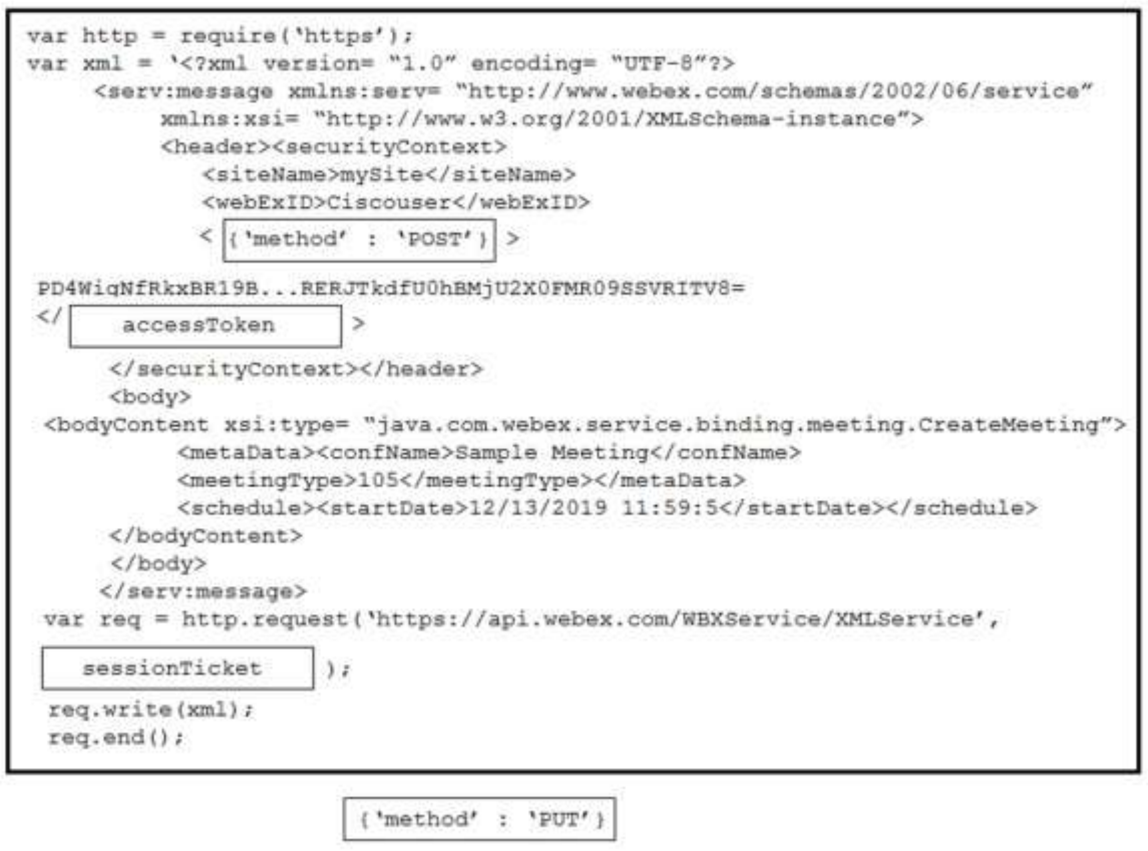
QUESTION 48
Drag and Drop Question
Drag and drop the code onto the snippet to construct the JavaScript to create a new meeting with the Webex Meetings XML API. Options can be used more than once.



```
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>
             <                    >
 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
```

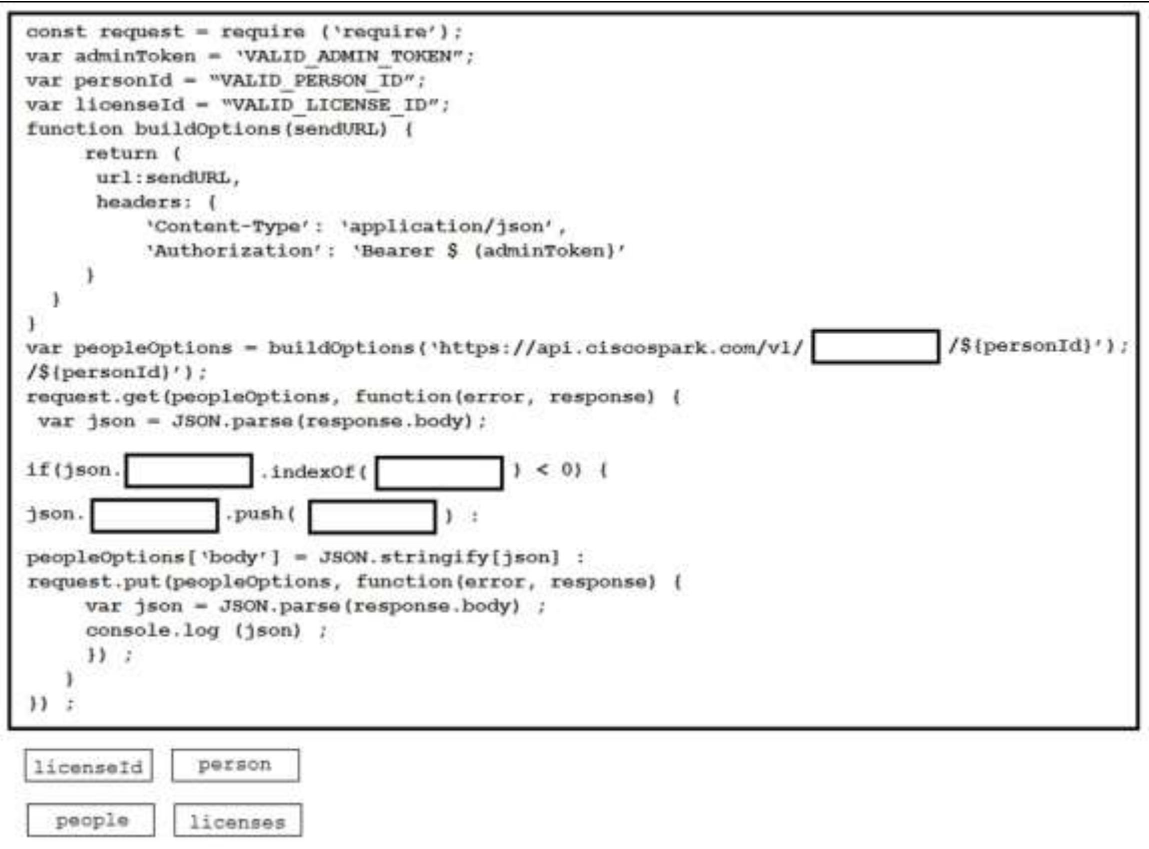
Answer:



```
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 49
Drag and Drop Question
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.

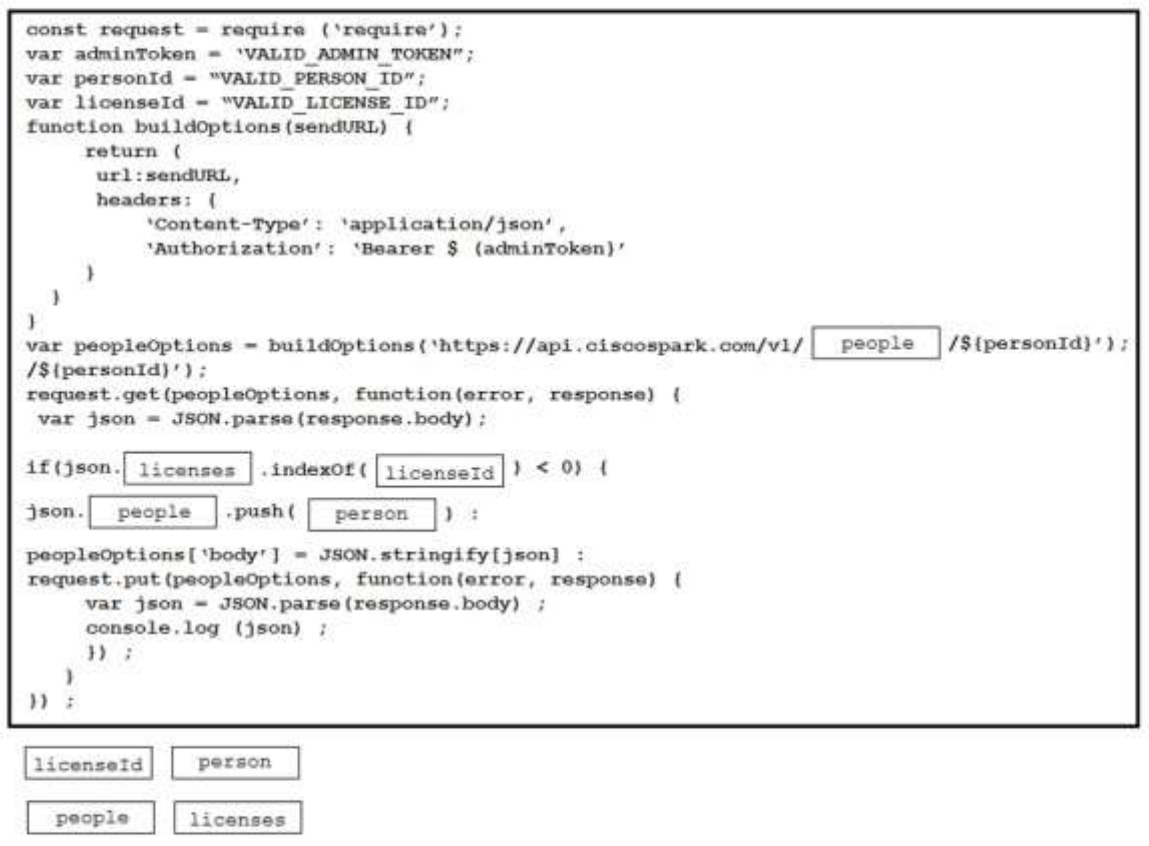


```
const request = require ('require');
var adminToken = 'VALID_ADMIN_TOKEN";
var personId = "VALID_PERSON_ID";
var licenseId = "VALID_LICENSE_ID";
function buildOptions(sendURL) {
     return (
      url:sendURL,
      headers: {
          'Content-Type': 'application/json',
          'Authorization':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



```
const request = require ('require');
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/[ people ]/${personId}');
/${personId}');
request.get(peopleOptions, function(error, response) {
 var json = JSON.parse(response.body);

if(json.[ licenses ].indexOf([ licenseId ]) < 0) {

json.[ people ].push([ person ]) ;

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

licenseId person

people licenses