

# CERTS LAND



## Cisco

### 300-920 Exam

Cisco Certified DevNet Professional

# Questions & Answers

(Demo Version - Limited Content)

Thank you for Downloading 300-920 exam PDF Demo

Get Full File:

<https://www.certsland.com/300-920-dumps/>

[www.certsland.com](https://www.certsland.com)

# Version: 4.0

---

**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.

<input type="text"/>	:	<input type="text"/>	<input type="text"/>
----------------------	---	----------------------	----------------------

admin:Password12345!	Authorization
Token	Bearer
BH782kjh-2jkhmnb3rlkj3lkjnm3kij3	Authentication
Basic	

---

**Answer:**

---

Authorization	:	Bearer	Token
---------------	---	--------	-------

Reference:

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

---

**Question: 2**

---

Refer to the exhibit.

```
const xml = `<?xml version= "1.0" encoding= "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>admin@cisco.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>3421</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?

A.

```
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);
});
```

B.

```
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. Option A
- B. Option B
- 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 roomkit devices.

---

**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</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',
  [redacted] );
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=
      </ [redacted] >
    </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',
  [redacted] );
req.write(xml);
req.end();

```

**Thank You for trying 300-920 PDF Demo**

<https://www.certsland.com/300-920-dumps/>

**Start Your 300-920 Preparation**

*[Limited Time Offer]* Use Coupon "**SAVE20**" for extra 20% discount on the purchase of PDF file. Test your 300-920 preparation with actual exam questions

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