



CHAPTER 4

User Resource Operations

A Cisco WebEx Social user is someone who is registered in the Cisco WebEx Social system. The Cisco WebEx Social API User resource operations let you obtain information about users and perform various operations for users.

[Table 4-1](#) provides a summary of the User resource operations. Each operation is described in detail in the section that is listed.

These sections provide this information:

- Purpose—Brief explanation of the operation
- Availability—Cisco WebEx Social releases that support the operation
- Request—HTTP method and URI for the operation
- Request Payload—Description of the request payload, if the operation requires one
- Response—Description of the expected response when the operation executes
- Examples—Sample requests and responses for the operation

In these sections, “you,” “your,” and “yourself” refer to the authenticated user who is executing an operation.

Table 4-1 *User Resource Operations Summary*

Operation and Reference	Purpose
Get Information about Yourself, page 4-5	Retrieves information about you
Get Information about a User, page 4-7	Retrieves information about the designated user
Get Information about Users, page 4-9	Retrieves information about the designated users
Get Information about a User with the Designated E-mail Address or Screen Name, page 4-11	Retrieves information about the user with the designated e-mail address or Cisco WebEx Social screen name
Update Information for a User, page 4-13	Updates information for the designated user
Get Information about Your Profile Images, page 4-16	Retrieves information about your Cisco WebEx Social profile images
Get Information about the Profile Images of a Designated User, page 4-18	Retrieves information about the Cisco WebEx Social profile images of the designated user
Upload Profile Image for Yourself, page 4-20	Uploads your Cisco WebEx Social profile image
Get Information about Communities to which You Belong, page 4-22	Retrieves information about communities to which you belong

Table 4-1 User Resource Operations Summary (continued)

Operation and Reference	Purpose
Get Information about Communities to which a User Belongs, page 4-24	Retrieves information about all communities to which the designated user belongs
Get Information about Communities that are Visible on Your Profile, page 4-26	Retrieves information about communities that are visible on your profile
Get Information about Communities that are Visible on the Profile of a User, page 4-28	Retrieves information about communities that are visible on the profile of the designated user
Get Information about Posts that are Visible on Your Profile, page 4-30	Retrieves information about all posts that are visible on your profile
Get Information about Posts that are in Your Library, page 4-32	Retrieves information about all posts that are in your library
Get Information about Recent Posts that are in Your Library, page 4-34	Retrieves information about posts that are in your library and that have been posted within the past 2 weeks, excluding posts that you made
Get Information about Posts that are Visible on the Profile of a User, page 4-36	Retrieves information about all posts that are visible on the profile of the designated user
Get Information about the Latest Micropost that You Made, page 4-38	Retrieves information about the latest micropost that you made
Get Information about the Latest Micropost that a User Made, page 4-40	Retrieves information about the latest micropost that the designated user made
Get Information about Posts that You have Marked as Favorites, page 4-42	Retrieves information about Favorite resources that have as the destination the posts that you have marked as favorites
Get Information about Posts that a User has Marked as Favorites, page 4-44	Retrieves information about Favorite resources that have as the destination the posts that the designated user has marked as favorites
Get Information about Your Document Library, page 4-46	Retrieves information about your document library
Get Information about the Document Library of a User, page 4-48	Retrieves information about the document library of a designated user
Get Information about Document Files in Your Document Library, page 4-50	Retrieves information about all documents that are in your document library
Get Information about Document Files in Your Document Library that are Authored by You, page 4-52	Retrieves information about document files that are in your document library and that are authored by you
Get Information about Document Files in Your Document Library that are Authored by a Designated User, page 4-54	Retrieves information about document files that are in your document library and that are authored by the designated user
Get Information about Document Files in the Document Library of a User, page 4-56	Retrieves information about all document files that are in the document library of the designated user
Get Information about Document Files in the Document Library of a User that are Authored by the Designated User, page 4-58	Retrieves information about document files that are in the document library of the designated user and that are authored by the designated user

Table 4-1 *User Resource Operations Summary (continued)*

Operation and Reference	Purpose
Get Information about the Designated Custom Attributes of a User, page 4-60	Retrieves information about the designated custom attributes of the designated user
Get Information about All Custom Attributes of a User, page 4-62	Retrieves information about all custom attributes of the designated user
Update Custom Attributes of a User, page 4-68	Updates designated custom attributes for the designated user
Get Information about All Activities that are Visible to You, page 4-71	Retrieves information about all activities that are visible to you in Cisco WebEx Social
Get Information about All Activities that Appear in Your Watch List, page 4-74	Retrieves information about all activities that appear in your Cisco WebEx Social watch list
Get Information about all Activities that Appear in the Profile Page of a User, page 4-76	Retrieves information about all activities that appear in the profile page of the designated user
Create a Favorite Relationship for Yourself, page 4-79	Creates a favorite relationship between you and a Cisco WebEx Social post
Remove a Favorite Relationship for Yourself, page 4-81	Removes the designated favorite relationship in which you favorited a Cisco WebEx Social post
Create a Follow Relationship for Yourself, page 4-82	Creates a follow relationship for yourself in which you are following another Cisco WebEx Social user
Get Information about Users that are Following You, page 4-84	Retrieves Follow resources that have as the destination the users that are following you
Get the Number of Users that are Following You, page 4-86	Retrieves the number of users that are following you
Get Information about Users that are Following the Designated User, page 4-87	Retrieves Follow resources that have as the destination the users that are following the designated user
Get the Number of Users that are Following the Designated User, page 4-89	Retrieves the number of users that are following the designated user
Get Information about Users that You are Following, page 4-90	Retrieves Follow resources that have as the destination the users that you are following
Get the Number of Users that You are Following, page 4-92	Retrieves the number of users that you are following
Get Information about Users that the Designated User is Following, page 4-93	Retrieves Follow resources that have as the destination the users that the designated user is following
Get the Number of Users that the Designated User is Following, page 4-95	Retrieves the number of users that the designated user is following
Remove Your Following of Another Cisco WebEx Social User, page 4-96	Remove the designated follow relationship in which you are following another Cisco WebEx Social user
Get Information about Users that You and the Designated User Both are Following, page 4-97	Retrieves Follow resources that have as the destination the users you and a designated user both are following

Table 4-1 *User Resource Operations Summary (continued)*

Operation and Reference	Purpose
Get the Number of Users that You and the Designated User Both are Following, page 4-99	Retrieves the number of users that you and a designated user both are following
Create a Like Relationship for Yourself, page 4-100	Creates a like relationship in which you like a Cisco WebEx Social post
Remove a Like Relationship for Yourself, page 4-102	Removes the designated like relationship in which you like a Cisco WebEx Social post

Get Information about Yourself

Purpose Retrieves information about you.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the User resource to include in a response ([Table 3-38 on page 3-34](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one User resource (the “[User Resource](#)” section on page 3-34 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the User resource. For an example, see the [“User Resource Response: XML Format, One Resource Returned”](#) section on page 16-2.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the User resource. For an example, see the [“User Resource Response: JSON Format, One Resource Returned”](#) section on page 16-13.

Get Information about a User

Purpose Retrieves information about the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	http://server[:port_number]/api/quad/rest/users/user_id[?query_parameters]

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user about whom to retrieve information. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the User resource to include in a response ([Table 3-38 on page 3-34](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one User resource (the [“User Resource” section on page 3-34](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/1234
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the User resource. For an example, see the [“User Resource Response: XML Format, One Resource Returned”](#) section on page 16-2.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/1234
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the User resource. For an example, see the [“User Resource Response: JSON Format, One Resource Returned”](#) section on page 16-13.

Get Information about Users

Purpose Retrieves information about the designated users.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id[,...[user_id]]</code> <code>[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of a user about whom to retrieve information. When specifying multiple identifiers, separate each one with a comma. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the User resource to include in a response ([Table 3-38 on page 3-34](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is not supported for paginating return results
 - The **count** query parameter is not supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more User resources (the [“User Resource” section on page 3-34](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/1234,9876
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the User resource. For an example, see the [“User Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-6.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234,9876
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the User resource. For an example, see the [“User Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-18.

Get Information about a User with the Designated E-mail Address or Screen Name

Purpose Retrieves information about the user with the designated e-mail address or Cisco WebEx Social screen name.

Availability Cisco WebEx Social 3.0 and later.

Request	HTTP Method	URI
	GET	<code>http://server[:port_number]/api/quad/rest/users?filterBy={email screenName}&filterValue=value[&query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *value*:
 - For **filterBy=email**, *value* is the e-mail address of the user about whom to retrieve information.
 - For **filterBy=screenName**, *value* is the Cisco WebEx Social screen name of the user about whom to retrieve information.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the User resource to include in a response ([Table 3-38 on page 3-34](#) describes these fields)
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one User resource (the “[User Resource](#)” section on page 3-34 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users?filterBy=screenName&filterValue=patkim
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the User resource. For an example, see the [“User Resource Response: XML Format, One Resource Returned”](#) section on page 16-2.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users?filterBy=screenName&filterValue=patkim
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the User resource. For an example, see the [“User Resource Response: JSON Format, One Resource Returned”](#) section on page 16-13.

Update Information for a User

Purpose Updates information for the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
PUT	<code>http://server[:port_number]/api/quad/rest/users/user_id</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to update information. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload

The request payload includes fields from the User resource ([Table 3-38 on page 3-34](#) describes these fields):

Required Fields	Optional Fields
—	emails endWorkHour firstName friendlyJobTitle lastName legalHold ¹ manager phones startWorkHour tags timeZone

1. Can be updated only by a user with the Cisco WebEx Social system administrator privilege.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only.

XML Example

Request

```
PUT http://webexsocialserver/api/quad/rest/users/10332
Accept: application/xml
Content-Type: application/xml
```

```

<user>
  <firstName>Pat</firstName>
  <lastName>Kim</lastName>
  <emails>
    <id>110004</id>
    <address>patkim@email.com</address>
    <primary>>false</primary>
  </emails>
  <phones>
    <id>110005</id>
    <primary>>false</primary>
    <number>8885556654</number>
    <extension></extension>
    <type>mobile</type>
  </phones>
  <endWorkHour>22</endWorkHour>
  <friendlyJobTitle>Software Tester</friendlyJobTitle>
  <startWorkHour>10</startWorkHour>
  <tags>
    <name>cooking</name>
  </tags>
  <tags>
    <name>reading</name>
  </tags>
  <timeZone>America/Los_Angeles</timeZone>
  <status>ACTIVE</status>
  <legalHold>NOHOLD</STATUS>
  <manager><id>10426</id></manager>
</user>

```

Response

HTTP/1.1 204 No Content

JSON Example**Request**

```

PUT http://webexsocialserver/api/quad/rest/users/10332
Accept:application/json
Content-Type: application/json

```

```

{
  "firstName": "pat",
  "lastName": "kim",
  "endWorkHour": 22,
  "friendlyJobTitle": "Software Developer",
  "startWorkHour": 10,
  "tags": [
    {
      "name": "quad",
    },
    {
      "name": "java"
    }
  ],
  "phones": [
    {
      "id": 110005,
      "primary": false,
      "number": "8885556654",
      "extension": "",
      "type": "mobile"
    }
  ]
}

```

```
    ],
    "manager": [
      {
        "id": 1234
      }
    ],
    "emails": [
      {
        "id": 110004,
        "resource": "http://quad.cisco.com/schema/1.0/email",
        "address": " patkim@email.com",
        "primary": false
      }
    ],
    "status": "ACTIVE",
    "legalHold": "NOHOLD",
    "timeZone": "America/Los_Angeles"
  }
}
```

Response

HTTP/1.1 204 No Content

Get Information about Your Profile Images

Purpose Retrieves information about your Cisco WebEx Social profile images.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/images/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the ProfileImage resource to include in a response ([Table 3-33 on page 3-32](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is not supported for paginating return results
 - The **count** query parameter is not supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more ProfileImage resources (the “[ProfileImage Resource](#)” section on page 3-32 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/images/@profile
Accept:application/xml
```


Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-154.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/images/@profile
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-156.

Get Information about the Profile Images of a Designated User

Purpose Retrieves information about the Cisco WebEx Social profile images of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/images/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about the profile image. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the ProfileImage resource to include in a response ([Table 3-33 on page 3-32](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is not supported for paginating return results
 - The **count** query parameter is not supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more ProfileImage resources (the [“ProfileImage Resource” section on page 3-32](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/1234/images/@profile
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-154.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/images/@profile
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-156.

Upload Profile Image for Yourself

Purpose

Uploads your Cisco WebEx Social profile image to the Cisco WebEx Social server. When you upload an image, the system automatically creates six versions of that image in these sizes:

- SMALL—24 x 24 pixels
- MEDIUM1—31 x 31 pixels
- MEDIUM2—48 x 48 pixels
- MEDIUM3—64 x 64 pixels
- MEDIUM4—110 x 110 pixels
- LARGE—150 x 150 pixels

Availability

Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
POST	<code>http://server[:port_number]/api/quad/rest/users/@me/images/@profile</code> <code>[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the ProfileImage resource to include in a response ([Table 3-33 on page 3-32](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload

Image to be uploaded.

The HTTP Content-Type header must be multipart/form-data for this request. The form-data key must be specified as `file=image_to_upload`.

Response

Standard response (see the [“API Responses”](#) section on page 1-15) with a response payload that can include one or more ProfileImage resources (the [“ProfileImage Resource”](#) section on page 3-32 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
POST http://quadserver]/api/quad/rest/users/@me/images/@profile
accept: application/xml
content-length: 87899

expect: 100-continue
content-type: multipart/form-data; boundary=-----02a4bb521c19
-----02a4bb521c19
Content-Disposition: form-data; name="file"; filename="ProfilePhoto.jpg"
<binary_file>
-----02a4bb521c19--
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-154.

JSON Example**Request**

```
POST http://quadserver]/api/quad/rest/users/@me/images/@profile
accept: application/json
content-length: 87899

expect: 100-continue
content-type: multipart/form-data; boundary=-----02a4bb521c19
-----02a4bb521c19
Content-Disposition: form-data; name="file"; filename="ProfilePhoto.jpg"
<binary_file>
-----02a4bb521c19--
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the ProfileImage resource. For an example, see the [“ProfileImage Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-156.

Get Information about Communities to which You Belong

Purpose Retrieves information about communities to which you belong.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/communities[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Community resource to include in a response ([Table 3-9 on page 3-10](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is supported for sorting resources in a response by the **name**, **stateChangedDate**, or **type** field in the Community resource
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Community resources (the “[Community Resource](#)” section on page 3-9 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/communities
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-92.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/communities
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-97.

Get Information about Communities to which a User Belongs

Purpose Retrieves information about all communities to which the designated user belongs. You must have the Cisco WebEx Social system administrator privilege to retrieve information for a user other than yourself.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/communities[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about belonged-to communities. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Community resource to include in a response ([Table 3-9 on page 3-10](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is supported for sorting resources in a response by the **name**, **stateChangedDate**, or **type** field in the Community resource
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more Community resources (the [“Community Resource” section on page 3-9](#) describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/communities
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-92.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/communities
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-97.

Get Information about Communities that are Visible on Your Profile

Purpose Retrieves information about communities that are visible on your profile.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/communities/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Community resource to include in a response ([Table 3-9 on page 3-10](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is not supported for paginating return results
 - The **count** query parameter is not supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Community resources (the “[Community Resource](#)” section on page 3-9 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/communities/@profile
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK  
Content-Type: application/xml
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-92.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/communities/@profile  
Accept:application/json
```

Response

```
HTTP/1.1 200 OK  
Content-Type: application/json
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-97.

Get Information about Communities that are Visible on the Profile of a User

Purpose Retrieves information about communities that are visible on the profile of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/communities/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about communities. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Community resource to include in a response ([Table 3-9 on page 3-10](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is not supported for paginating return results
 - The **count** query parameter is not supported for designating the maximum number of items in the return results
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more Community resources (the [“Community Resource” section on page 3-9](#) describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/communities/@profile
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-92.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/communities/@profile
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Community resource. For an example, see the [“Community Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-97.

Get Information about Posts that are Visible on Your Profile

Purpose Retrieves information about all posts that are visible on your profile.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/posts/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response (Table 3-31 on page 3-28 describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Post resources (the “[Post Resource](#)” section on page 3-28 describes this resource). XML, JSON, and Atom formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@profile
Accept: application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-118.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@profile
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-137.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@profile
Accept: application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-143.

Get Information about Posts that are in Your Library

Purpose Retrieves information about all posts that are in your library.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/{@meluser_id}/posts[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Your user identifier. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response ([Table 3-31 on page 3-28](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is supported for sorting resources in a response by the **creator**, **lastModifiedDate**, or **title** field in the Post resource
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more Post resources (the [“Post Resource” section on page 3-28](#) describes this resource). XML, JSON, and Atom formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts
Accept:application/xml
```


Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-118.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-137.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts
Accept:application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-143.

Get Information about Recent Posts that are in Your Library

Purpose Retrieves information about recent posts that are in your library, excluding posts that you made. “Recent” means posts that were added or edited within the past 2 weeks.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/{@meuser_id}/posts/@recent[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Your user identifier. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response ([Table 3-31 on page 3-28](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is supported for sorting resources in a response by the **creator**, **lastModifiedDate**, or **title** field in the Post resource
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more Post resources (the [“Post Resource” section on page 3-28](#) describes this resource). XML, JSON, and Atom formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@recent
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-118.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@recent
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-137.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@recent
Accept:application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-143.

Get Information about Posts that are Visible on the Profile of a User

Purpose Retrieves information about all posts that are visible on the profile of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/posts/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve posts. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response ([Table 3-31 on page 3-28](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Post resources (the “[Post Resource](#)” section on page 3-28 describes this resource). XML, JSON, and Atom formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/posts/@profile
Accept: application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-118.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/posts/@profile
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-137.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/posts/@profile
Accept: application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-143.

Get Information about the Latest Micropost that You Made

Purpose Retrieves information about the latest micropost that you made.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/posts/@status[?query_parameters]</code>

Where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response (Table 3-31 on page 3-28 describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one Post resource (the “[Post Resource](#)” section on page 3-28 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@status
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-118.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/posts/@status
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-137.

Get Information about the Latest Micropost that a User Made

Purpose Retrieves information about the latest micropost that the designated user made.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/posts/@status[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user from whom to retrieve the latest micropost. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Post resource to include in a response ([Table 3-31 on page 3-28](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one Post resource (the [“Post Resource” section on page 3-28](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/12345/posts/@status
Accept:application/xml
```


Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: XML Format, One Resource Returned”](#) section on page 16-111.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/12345/posts/@status
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Post resource. For an example, see the [“Post Resource Response: JSON Format, One Resource Returned”](#) section on page 16-131.

Get Information about Posts that You have Marked as Favorites

Purpose Retrieves information about Favorite resources that have as the destination the posts that you have marked as favorites.

Availability Cisco WebEx Social 3.0 and later.

Request	HTTP Method	URI
	GET	<code>http://server[:port_number]/api/quad/rest/users/@me/favorite/@favoritingposts[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Favorite resource to include in a response ([Table 3-22 on page 3-22](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Favorite resources (the “[Favorite Relationship Resource](#)” section on page 3-22 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://quadserver/api/quad/rest/users/@me/favorite
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: XML Format, Multiple Resources Returned”](#) section on page 16-164.

JSON Example**Request**

```
GET http://quadserver/aapi/quad/rest/users/@me/favorite
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: JSON Format, Multiple Resources Returned”](#) section on page 16-182.

Get Information about Posts that a User has Marked as Favorites

Purpose Retrieves information about Favorite resources that have as the destination the posts that the designated user has marked as favorites.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/favorite/@favoritingposts[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about posts that the user marked as favorites. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Favorite resource to include in a response ([Table 3-22 on page 3-22](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter is not supported for sorting resources in a response
 - The **startIndex** query parameter is supported for paginating return results
 - The **count** query parameter is supported for designating the maximum number of items in the return results
 - The **meta** query parameter is supported with the **like** and the **favorite** values for requesting relationship metadata

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Favorite resources (the “[Favorite Relationship Resource](#)” section on page 3-22 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/favorite
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: XML Format, Multiple Resources Returned”](#) section on page 16-164.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/favorite
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: JSON Format, Multiple Resources Returned”](#) section on page 16-182.

Get Information about Your Document Library

Purpose Retrieves information about your document library.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/documentlibrary[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response (Table 3-19 on page 3-20 describes these fields).
 - The **filterBy** query parameter is not supported for filtering resources in a response.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one DocumentLibrary resource that contains representations of top-level document folders (the “[DocumentLibrary Resource](#)” section on page 3-20 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned”](#) section on page 16-151.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned”](#) section on page 16-152.

Get Information about the Document Library of a User

Purpose Retrieves information about the document library of a designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/documentlibrary[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve document library information. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response ([Table 3-19 on page 3-20](#) describes these fields).
 - The **filterBy** query parameter is not supported for filtering resources in a response.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one DocumentLibrary resource that contains representations of top-level document folders (the [“DocumentLibrary Resource” section on page 3-20](#) describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users1234/documentlibrary
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned”](#) section on page 16-151.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users1234/documentlibrary
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned”](#) section on page 16-152.

Get Information about Document Files in Your Document Library

Purpose Retrieves information about all document files that are in your document library.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/documentlibrary/@recent[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response (Table 3-19 on page 3-20 describes these fields).
 - The **filterBy** query parameter is not supported for filtering resources in a response.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one DocumentLibrary resource that contains representations of all document files in your document library (the “[DocumentLibrary Resource](#)” section on page 3-20 describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary/@recent
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned”](#) section on page 16-151.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary/@recent
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned”](#) section on page 16-152.

Get Information about Document Files in Your Document Library that are Authored by You

Purpose Retrieves information about document files that are in your document library and that are authored by you.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/documentlibrary? ?filterBy=author&filterValue=@me[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response ([Table 3-19 on page 3-20](#) describes these fields).
 - The **filterBy** query parameter is required as shown in the URI.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one DocumentLibrary resource that contains representations of document files that are authored by you (the “[DocumentLibrary Resource](#)” section on page 3-20 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary?filterBy=author
&filterValue=@me
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned”](#) section on page 16-151.

JSON Example**Request**

```
http://webexsocialserver/api/quad/rest/users/@me/documentlibrary?filterBy=author
&filterValue=@me
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned”](#) section on page 16-152.

Get Information about Document Files in Your Document Library that are Authored by a Designated User

Purpose Retrieves information about document files that are in your document library and that are authored by the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/documentlibrary?filterBy=author&filterValue=user_id[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user who authored the document files. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on [page 1-11](#).
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on [page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response ([Table 3-19 on page 3-20](#) describes these fields).
 - The **filterBy** query parameter is required as shown in the URI.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one DocumentLibrary resource that contains representations of document files that are authored by the designated user (the [“DocumentLibrary Resource” section on page 3-20](#) describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary?filterBy=author
&filterValue=1234
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned” section on page 16-151](#).

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/documentlibrary?filterBy=author
&filterValue=1234
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned” section on page 16-152](#).

Get Information about Document Files in the Document Library of a User

Purpose Retrieves information about all document files that are in the document library of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/documentlibrary/@recent[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user whose document library contains the document files about which information is to be retrieved. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities”](#) section on page 1-11.
- *query_parameters*—Query parameters. See the [“Query Parameters”](#) section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response (Table 3-19 on page 3-20 describes these fields).
 - The **filterBy** query parameter is not supported for filtering resources in a response.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses”](#) section on page 1-15) with a response payload that can include one DocumentLibrary resource that contains representations of all document files in the document library or the designated user (the [“DocumentLibrary Resource”](#) section on page 3-20 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/documentlibrary/@recent
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned”](#) section on page 16-151.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/documentlibrary/@recent
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned”](#) section on page 16-152.

Get Information about Document Files in the Document Library of a User that are Authored by the Designated User

Purpose Retrieves information about document files that are in the document library of the designated user and that are authored by the designated user. These users can be the same or different people.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id_(library)/documentlibrary?filterBy=author&filterValue=user_id_(author)[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id_(library)*—Identifier of the user whose document library that contains the document files about which information is to be retrieved. For information about obtaining the identifier that you need, see the instructions for obtaining a *user_id* in [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *user_id_(author)*—Identifier of the user who authored the document files about which information is to be retrieved. For information about obtaining the identifier that you need, see the instructions for obtaining a *user_id* in [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the DocumentLibrary resource to include in a response ([Table 3-19 on page 3-20](#) describes these fields).
 - The **filterBy** query parameter is required as shown in the URI.
 - The **sortOrder** query parameter does not apply.
 - The **startIndex** query parameter is supported for paginating return results by designating the first DocumentFolder or DocumentFile child resource that is returned. The DocumentFolder child resource takes precedence.
 - The **count** query parameter is supported for designating the maximum number of DocumentFolder and DocumentFile child resources that are included in the return results. DocumentFolder child resources take precedence.
 - The **meta** query parameter does not apply.

Request Payload A request payload is not required.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one DocumentLibrary resource that contains representations of document files that are authored by the designated user (the [“DocumentLibrary Resource” section on page 3-20](#) describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/documentlibrary
?filterBy=author&filterValue=9876
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: XML Format, One Resource Returned” section on page 16-151](#).

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/documentlibrary
?filterBy=author&filterValue=9876
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the DocumentLibrary resource. For an example, see the [“DocumentLibrary Resource Response: JSON Format, One Resource Returned” section on page 16-152](#).

Get Information about the Designated Custom Attributes of a User

Purpose Retrieves information about the designated custom attributes of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request	HTTP Method	URI
	GET	http://server[:port_number]/api/quad/rest/users/user_id/customattributes/attributes

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about designated custom attributes. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *attributes*—Comma-separated string of custom attributes about which to retrieve information. To retrieve the custom attributes that you can designate here, use the operation that the [“Get Information about All Custom Attributes of a User” section on page 4-62](#) describes.

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more CustomAttribute resources (the [“CustomAttribute Resource” section on page 3-16](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/110006/customattributes/DOB,LOCATION
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:response xmlns:ns2="http://quad.cisco.com/ns/1.0/xml">
  <statusCode>200</statusCode>
  <startIndex>0</startIndex>
```

```

<itemsPerPage>2</itemsPerPage>
<totalResults>2</totalResults>
<filtered>>false</filtered>
<sorted>>false</sorted>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>DOB</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value>01/01/1980</value>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>LOCATION</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value>NewLocation</value>
  </customAttribute>
</entry>
</ns2:response>

```

JSON Example**Request**

```

GET http://webexsocialserver/api/quad/rest/users/110006/customattributes/DOB, LOCATION
Accept: application/json

```

Response

```

HTTP/1.1 200 OK
Content-Type: application/json

```

```

{ "entry" : [ { "data" : { "dataSelectionValues" : [ ],
  "dataType" : "String",
  "id" : 0,
  "key" : "DOB",
  "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
  "value" : "01/01/1980"
} },
  { "data" : { "dataSelectionValues" : [ ],
  "dataType" : "String",
  "id" : 0,
  "key" : "LOCATION",
  "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
  "value" : "NewLocation"
} }
],
  "filtered" : false,
  "itemsPerPage" : 2,
  "serverMessages" : null,
  "sorted" : false,
  "startIndex" : 0,
  "statusCode" : 200,
  "totalResults" : 2
}

```

Get Information about All Custom Attributes of a User

Purpose Retrieves information about all custom attributes of the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/customattributes</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about all custom attributes. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload

A request payload is not required.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that can include one or more CustomAttribute resources (the [“CustomAttribute Resource” section on page 3-16](#) describes this resource). XML and JSON formats are supported.

XML Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/1234/customattributes
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:response xmlns:ns2="http://quad.cisco.com/ns/1.0/xml">
  <statusCode>200</statusCode>
  <startIndex>0</startIndex>
  <itemsPerPage>17</itemsPerPage>
  <totalResults>17</totalResults>
  <filtered>>false</filtered>
  <sorted>>false</sorted>
  <entry>
    <customAttribute>
      <dataType>String[]</dataType>
      <id>0</id>
```

```

        <key>text</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">12345678</dataSelectionValues>
        <dataType>int[]</dataType>
        <id>0</id>
        <key>multivalue</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>String[]</dataType>
        <id>0</id>
        <key>test</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">12</dataSelectionValues>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">23</dataSelectionValues>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">45</dataSelectionValues>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">56</dataSelectionValues>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">67</dataSelectionValues>
        <dataType>String[]</dataType>
        <id>0</id>
        <key>selectiontextvalues</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>String</dataType>
        <id>0</id>
        <key>WEBPHONE_KEY_LAST_PHONE_MODE</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>int[]</dataType>
        <id>0</id>
        <key>multiple int</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>

```

```

    <customAttribute>
      <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">12</dataSelectionValues>
      <dataType>int[]</dataType>
      <id>0</id>
      <key>dropdown</key>
      <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
      <value/>
    </customAttribute>
  </entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>WEBPHONE_KEY_CTI_PASSWORD</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value/>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>WEBPHONE_KEY_LAST_SUCCESSFUL_CTI</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value/>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>WEBPHONE_KEY_LAST_SUCCESSFUL_TFTP</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value/>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>multipleString</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value/>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataType>String</dataType>
    <id>0</id>
    <key>WEBPHONE_KEY_ACCEPT_VIDEO_CALLS</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
    <value/>
  </customAttribute>
</entry>
<entry>
  <customAttribute>
    <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">1221111</dataSelectionValues>
    <dataType>int[]</dataType>
    <id>0</id>
    <key>selectioninteger</key>
    <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>

```



```

        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataSelectionValues xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">123</dataSelectionValues>
        <dataType>int[]</dataType>
        <id>0</id>
        <key>PresetIntegerValues</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>String</dataType>
        <id>0</id>
        <key>WEBPHONE_KEY_SUCCESS_CTI_ADDRESS</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>String</dataType>
        <id>0</id>
        <key>WEBPHONE_KEY_LAST_DEVICE_NAME</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
<entry>
    <customAttribute>
        <dataType>String</dataType>
        <id>0</id>
        <key>WEBPHONE_KEY_LAST_LINE_DN</key>
        <resource>http://quad.cisco.com/schema/1.0/customattribute</resource>
        <value/>
    </customAttribute>
</entry>
</ns2:response>

```

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/customattributes
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{ "entry" : [ { "data" : { "dataSelectionValues" : [ ],
    "dataType" : "String[]",
    "id" : 0,
    "key" : "text",
    "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
    "value" : ""
  } },
  { "data" : { "dataSelectionValues" : [ "12345678" ],
    "dataType" : "int[]",
```

```

        "id" : 0,
        "key" : "multivalue",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "String[]",
        "id" : 0,
        "key" : "test",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ "12",
        "23",
        "45",
        "56",
        "67"
    ],
        "dataType" : "String[]",
        "id" : 0,
        "key" : "selectiontextvalues",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "String",
        "id" : 0,
        "key" : "WEBPHONE_KEY_LAST_PHONE_MODE",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "int[]",
        "id" : 0,
        "key" : "multiple int",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ "12" ],
        "dataType" : "int[]",
        "id" : 0,
        "key" : "dropdown",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "String",
        "id" : 0,
        "key" : "WEBPHONE_KEY_CTI_PASSWORD",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "String",
        "id" : 0,
        "key" : "WEBPHONE_KEY_LAST_SUCCESSFUL_CTI",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
        "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
        "dataType" : "String",
        "id" : 0,
        "key" : "WEBPHONE_KEY_LAST_SUCCESSFUL_TFTP",
        "resource" : "http://quad.cisco.com/schema/1.0/customattribute",

```

```

        "value" : ""
      } },
    { "data" : { "dataSelectionValues" : [ ],
      "dataType" : "String",
      "id" : 0,
      "key" : "multipleString",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
      "dataType" : "String",
      "id" : 0,
      "key" : "WEBPHONE_KEY_ACCEPT_VIDEO_CALLS",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ "1221111" ],
      "dataType" : "int[]",
      "id" : 0,
      "key" : "selectioninteger",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ "123" ],
      "dataType" : "int[]",
      "id" : 0,
      "key" : "PresetIntegerValues",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
      "dataType" : "String",
      "id" : 0,
      "key" : "WEBPHONE_KEY_SUCCESS_CTI_ADDRESS",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
      "dataType" : "String",
      "id" : 0,
      "key" : "WEBPHONE_KEY_LAST_DEVICE_NAME",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } },
    { "data" : { "dataSelectionValues" : [ ],
      "dataType" : "String",
      "id" : 0,
      "key" : "WEBPHONE_KEY_LAST_LINE_DN",
      "resource" : "http://quad.cisco.com/schema/1.0/customattribute",
      "value" : ""
    } }
  ],
  "filtered" : false,
  "itemsPerPage" : 17,
  "serverMessages" : null,
  "sorted" : false,
  "startIndex" : 0,
  "statusCode" : 200,
  "totalResults" : 17
}

```

Update Custom Attributes of a User

Purpose Updates designated custom attributes for the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
PUT	<code>http://server[:port_number]/api/quad/rest/users/user_id/customattributes/attributes</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to update custom attributes. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *attributes*—Comma-separated string of custom attributes to be updated. To retrieve the custom attributes that you can designate here, use the operation that the [“Get Information about All Custom Attributes of a User” section on page 4-62](#) describes.

Request Payload

The request payload includes the following fields from the CustomAttribute resource ([Table 3-15 on page 3-17](#) describes these fields). You must include one pair of **key** and **value** fields for each custom attribute that you specify in the request. These **key/value** pairs must be included in the same order as their corresponding custom attributes are specified in the request.

Required Fields	Optional Fields
key value	—

Response

When updating a single value, standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only.

When updating multiple values, standard response with a payload that includes the status of the updates. XML and JSON formats are supported.

XML Example

Request—Single Custom Attribute Updated

```
PUT http://webexsocialserver/api/quad/rest/users/110006/customattributes/DOB
Accept:application/xml
Content-Type: application/xml
```

```

<customAttributes>
  <customAttribute>
    <key>DOB</key>
    <value>01/01/1980</value>
  </customAttribute>
</customAttributes>

```

Response—Single Custom Attribute Updated

```

HTTP/1.1 204 No Content
Content-Type: application/xml

```

Request—Multiple Custom Attributes Updated

```

PUT http://webexsocialserver/api/quad/rest/users/110006/customattributes/DOB,LOCATION
Accept:application/xml
Content-Type: application/xml

```

```

<customAttributes>
  <customAttribute>
    <key>DOB</key>
    <value>01/01/1980</value>
  </customAttribute>
  <customAttribute>
    <key>LOCATION</key>
    <value>NewLocation</value>
  </customAttribute>
</customAttributes>

```

Response—Multiple Custom Attributes

```

HTTP/1.1 207 Multi-Status
Content-Type: application/xml

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:response xmlns:ns2="http://quad.cisco.com/ns/1.0/xml">
  <statusCode>200</statusCode>
  <startIndex>0</startIndex>
  <itemsPerPage>0</itemsPerPage>
  <totalResults>0</totalResults>
  <filtered>>false</filtered>
  <sorted>>false</sorted>
  <serverMessages>
    <code>QUAD-600</code>
    <description>Success</description>
    <message>Custom attribute with key [ DOB ] is updated successfully.</message>
  </serverMessages>
  <serverMessages>
    <code>QUAD-600</code>
    <description>Success</description>
    <message>Custom attribute with key [ LOCATION ] is updated successfully.</message>
  </serverMessages>
</ns2:response>

```

JSON Example

Request—Single Custom Attribute Updated

```

PUT http://webexsocialserver/api/quad/rest/users/110006/customattributes/DOB
Accept:application/json
Content-Type: application/json

```

```

Payload:
{ "customAttribute" : [ { "key" : "DOB",

```

```

        "value" : "01/01/1980"
      }
    ] }

```

Response—Single Custom Attribute Updated

HTTP/1.1 204 No Content
Content-Type: application/json

Request—Multiple Custom Attributes Updated

PUT http://webexsocialserver/api/quad/rest/users/5710001/customattributes/DOB,LOCATION
Accept:application/json
Content-Type: application/json

```

{ "customAttribute" : [ { "key" : "DOB",
                        "value" : "01/01/1980"
                      },
  { "key" : "LOCATION",
    "value" : "NewLocation"
  }
] }

```

Response—Multiple Custom Attributes Updated

HTTP/1.1 207 Multi-Status
Content-Type: application/json

```

{ "entry" : [ ],
  "filtered" : false,
  "itemsPerPage" : 0,
  "serverMessages" : [ { "code" : "QUAD-600",
                        "description" : "Success",
                        "message" : "Custom attribute with key [ DOB ] is updated successfully."
                      },
  { "code" : "QUAD-600",
    "description" : "Success",
    "message" : "Custom attribute with key [ LOCATION ] is updated successfully."
  }
],
  "sorted" : false,
  "startIndex" : 0,
  "statusCode" : 200,
  "totalResults" : 0
}

```

Get Information about All Activities that are Visible to You

Purpose

Retrieves information about all activities that are visible to you in Cisco WebEx Social.

An activity consists of these components: *actor*, *action*, *object*, and *target*. An activity describes an action that is performed by an actor with or on an object that relates to a target. For example, an actor could be a Cisco WebEx Social user, an action could be “created,” an object could be a document, and a target could be a library.

A component may be explicit or implied. For example, an activity could be described as “Nicole updated a document in her library” or “Mike shared a post.”

Availability

Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/{@me user_id}/activities[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Your user identifier. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter does not apply.
 - The **filterBy** query parameter is supported in this format:
filterBy=@user|@community|@tag|@activityType}&filterValue=value.

For **filterBy=@user**, *value* is the identifier of a Cisco WebEx Social user. In this case, the operation retrieves only activities that are visible to you and that include this user. You can use a *value* of **@me** to designate yourself. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.

For **filterBy=@community**, *value* is the identifier of a Cisco WebEx Social community. In this case, the operation retrieves only activities that are visible to you and that include this community. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.

For **filterBy=@tag**, *value* is the name of a tag to an entity. In this case, the operation retrieves only activities that are visible to you and that include this tag.

For **filterBy=@activitytype**, *value* is one or more of the following tags. If you use multiple tags, separate each one with a comma followed by a space.

- **@comment**—Causes the operation to retrieve only activities that include comments
- **@create**—Causes the operation to retrieve only activities that include creating a post, micropost, user wall post, or community wall post
- **@edit**—Causes the operation to retrieve only activities that editing a post, micropost, user wall post, or community wall post
- **@like**—Causes the operation to retrieve only activities that are generated by a user liking any content
- **@mutualfollow**—Causes the operation to retrieve only activities that are generated when a user enters into a mutual follow relationship with another user
- **@share**—Causes the operation to retrieve only activities that are generated when a post is shared with a community
- The **sortOrder** query parameter is not supported for sorting resources in a response.
- The **startIndex** query parameter is not supported for paginating return results.
- The **count** query parameter is supported for designating the maximum number of items in the return results. The system retrieves activities in reverse chronological order, stopping when it retrieves the designated maximum number of activities. (If you specify the **count** and the **timestamp** query parameters in the same request, the system stops retrieving activities when it retrieves the designated maximum number of activities or when it reaches the *time_since* date and time, whichever occurs first.)
- The **meta** query parameter does not apply.
- The special query parameter **timestamp=time_since** is supported for retrieving activities that occurred between the date and time that you designate and the date and time that you execute this operation. Replace *time_since* with the date and time, in UTC format, from which no earlier activities are to be retrieved.

Request Payload

A request payload is not required.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that includes the following:

Item	Explanation
objectType	Appears as collection .
timestamp	Date and time that the earliest activity that is returned occurred, in UTC format.
totalItems	Total number of items that are returned.
items	Array of Activity activities resources. The “ Activity Activities Resource ” section on page 3-38 describes this resource.

JSON Example

Request

```
GET http://webexsocialserver/api/quad/rest/users/210018/activities
Accept:application/json
```


Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Collection activities resource. For an example, see the [“Collection Activities Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-234.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/210018/activities
Accept: application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Collection activities resource. For an example, see the [“Collection Activities Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-241.

Get Information about All Activities that Appear in Your Watch List

Purpose

Retrieves information about all activities that appear in your Cisco WebEx Social watch list.

An activity consists of these components: *actor*, *action*, *object*, and *target*. An activity describes an action that is performed by an actor with or on an object that relates to a target. For example, an actor could be a Cisco WebEx Social user, an action could be “created,” an object could be a document, and a target could be a library.

A component may be explicit or implied. For example, an activity could be described as “Nicole updated a document in her library” or “Mike shared a post.”

Availability

Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/{@me user_id}/watchlist[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Your user identifier. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter does not apply.
 - The **filterBy** query parameter is not supported for filtering resources in a response.
 - The **sortOrder** query parameter is not supported for sorting resources in a response.
 - The **startIndex** query parameter is not supported for paginating return results.
 - The **count** query parameter is supported for designating the maximum number of items in the return results. The system retrieves activities in reverse chronological order.
 - The **meta** query parameter does not apply.

Request Payload

A request payload is not required.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with a response payload that includes the following:

Item	Explanation
objectType	Appears as .
timestamp	Date and time that the earliest activity that is returned occurred, in UTC format.
totalItems	Total number of items that are returned.
items	Array of Activity activities resources. The “Activity Activities Resource” section on page 3-38 describes this resource.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/210018/watchlist
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Collection activities resource. For an example, see the [“Collection Activities Resource Response: JSON Format, Multiple Resources Returned” section on page 16-234](#).

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/210018/watchlist
Accept: application/atom+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml
```

The response can also include the Collection activities resource. For an example, see the [“Collection Activities Resource Response: Atom Format, Multiple Resources Returned” section on page 16-241](#).

Get Information about all Activities that Appear in the Profile Page of a User

Purpose

Retrieves information about all activities that appear in the profile page of the designated user.

An activity consists of these components: *actor*, *action*, *object*, and *target*. An activity describes an action that is performed by an actor with or on an object that relates to a target. For example, an actor could be a Cisco WebEx Social user, an action could be “created,” an object could be a document, and a target could be a library.

A component may be explicit or implied. For example, an activity could be described as “Nicole updated a document in her library” or “Mike shared a post.”

Availability

Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/{@meluser_id}/activities/@profile[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve information about activities. You can use your identifier or **@me** to indicate yourself. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter does not apply.
 - The **filterBy** query parameter is supported in this format: **filterBy={@user|@community|@tag}&filterValue=value**.

For **filterBy=@user**, *value* is the identifier of a Cisco WebEx Social user. In this case, the operation retrieves only activities that include this user. You can use a *value* of **@me** to designate yourself. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.

For **filterBy=@community**, *value* is the identifier of a Cisco WebEx Social community. In this case, the operation retrieves only activities that include this community. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.

For **filterBy=@tag**, *value* is the name of a tag to an entity. In this case, the operation retrieves only activities that include this tag.

- The **sortOrder** query parameter is not supported for sorting resources in a response.

- The **startIndex** query parameter is not supported for paginating return results.
- The **count** query parameter is supported for designating the maximum number of items in the return results. The system retrieves activities in reverse chronological order, stopping when it retrieves the designated maximum number of activities. (If you specify the **count** and the **timestamp** query parameters in the same request, the system stops retrieving activities when it retrieves the designated maximum number of activities or when it reaches the *time_since* date and time, whichever occurs first.)
- The **meta** query parameter does not apply.
- The special query parameter **timestamp=time_since** is supported for retrieving activities that occurred between the date and time that you designate and the date and time that you execute this operation. Replace *time_since* with the date and time, in UTC format, from which no earlier activities are to be retrieved.

Request Payload

A request payload is not required.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that includes the following:

Item	Explanation
objectType	Appears as collection .
timestamp	Date and time that the earliest activity that is returned occurred, in UTC format.
totalItems	Total number of items that are returned.
items	Array of Activity activities resources. The “ Activity Activities Resource ” section on page 3-38 describes this resource.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/210018/activities/@profile
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Collection activities resource. For an example, see the “[Collection Activities Resource Response: JSON Format, Multiple Resources Returned](#)” section on page 16-234.

Atom Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/210018/activities/@profile
Accept: application/atom+xml
```

Response

HTTP/1.1 200 OK

Content-Type: application/atom+xml

The response can also include the Collection activities resource. For an example, see the [“Collection Activities Resource Response: Atom Format, Multiple Resources Returned”](#) section on page 16-241.

Create a Favorite Relationship for Yourself

Purpose Creates a favorite relationship between you and a Cisco WebEx Social post.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
POST	<code>http://server[:port_number]/api/quad/rest/users/@me/favorite</code> <code>[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Favorite resource to include in a response ([Table 3-22 on page 3-22](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload

The request payload includes fields from the Favorite resource ([Table 3-22 on page 3-22](#) describes these fields):

Required Fields	Optional Fields
destination ¹	—

1. The **destination** field defines a resource that represents the post with which you have the favorite relationship. The **id** and the **resource** fields in this resource are required. Other fields are not used.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one Favorite resource (the “[Favorite Relationship Resource](#)” section on page 3-22 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
POST http://webexsocialserver/api/quad/rest/users/@me/favorite
Accept:application/xml
content-type: xml

<favorite
  <destination>
    <resource>http://quad.cisco.com/schema/1.0/post</resource>
    <id>123</id>
  </destination>
</favorite>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://webexsocialserver/api/quad/rest/users/@me/favorite/user:1234,post:500001
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: XML Format, One Resource Returned”](#) section on page 16-158.

JSON Example**Request**

```
POST http://webexsocialserver/api/quad/rest/users/@me/favorite
Accept:application/json
content-type: json

{  "destination": {
    "id": "123",
    "resource": "http://quad.cisco.com/schema/1.0/post"
  }
}
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://webexsocialserver/api/quad/rest/users/@me/favorite/user:1234,post:500001
```

The response can also include the Favorite resource. For an example, see the [“Favorite Resource Response with Post Resource in Destination Field: JSON Format, One Resource Returned”](#) section on page 16-176.

Remove a Favorite Relationship for Yourself

Purpose Removes the designated favorite relationship in which you favorited a Cisco WebEx Social post.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
DELETE	<code>http://server[:port_number]/api/quad/rest/users/@me/favorite/favorite_id</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *favorite_id*—Identifier of the Favorite resource that defines the relationship to remove. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only.

Example

Request

```
DELETE http://webexsocialserver/api/quad/rest/users/@me/favorite/user:1234,post:500001
```

Response

```
HTTP/1.1 204 No Content
```

Create a Follow Relationship for Yourself

Purpose Creates a follow relationship for yourself in which you are following another Cisco WebEx Social user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
POST	http://server[:port_number]/api/quad/rest/users/@me/follow[?query_parameters]

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response ([Table 3-24 on page 3-24](#) describes these fields)
 - The **filterBy** query parameter is not supported for filtering resources in a response
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload

The request payload includes fields from the Follow resource ([Table 3-24 on page 3-24](#) describes these fields):

Required Fields	Optional Fields
destination ¹	—

1. The **destination** field defines a resource that represents the user that you are following. The **id** and the **resource** fields in this resource are required. Other fields are not used.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one Follow resource (the “[Follow Relationship Resource](#)” section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
POST http://webexsocialserver/api/quad/rest/users/@me/follow
Accept:application/xml
Content-Type:application/xml
```

```
<follow>
  <destination>
    <resource>http://quad.cisco.com/schema/1.0/user</resource>
    <id>123</id>
  </destination>
</follow>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://webexsocialserver/api/quad/rest/users/@me/follow/user:1234,user:9876
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, One Resource Returned”](#) section on page 16-195.

JSON Example**Request**

```
POST http://webexsocialserver/api/quad/rest/users/@me/follow
Accept:application/json
Content-Type:application/json
```

```
{
  {
    "destination": {
      "id": "123",
      "resource": "http://quad.cisco.com/schema/1.0/user"
    }
  }
}
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://webexsocialserver/api/quad/rest/users/@me/follow/user:1234,user:9876
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, One Resource Returned”](#) section on page 16-203.

Get Information about Users that are Following You

Purpose Retrieves Follow resources that have as the destination the users that are following you.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/follow/@followersusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
 - *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for detailed information about using these parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response (Table 3-24 on page 3-24 describes these fields).
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource.
 - The **sortOrder** query parameter is supported for sorting resources in a response when the **filterBy** query parameter is not used. In this case, resources can be sorted by the **firstName** or **lastName** field in the User resource, or by the special **timeStamp** field, which designates when the follow relationship was created.

The **sortOrder** query parameter is not supported for sorting resources in a response when the **filterBy** query parameter is used. In this case, resources are sorted in ascending order by the **fullName** field in the User resource.
 - The **startIndex** query parameter is supported for paginating return results.
 - The **count** query parameter is supported for designating the maximum number of items in the return results.
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata.

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Follow resources (the “[Follow Relationship Resource](#)” section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/follow/@followersusers
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-198.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/follow/@followersusers
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-206.

Get the Number of Users that are Following You

Purpose Retrieves the number of the users that are following you.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
HEAD	<code>http://server[:port_number]/api/quad/rest/users/@me/follow/@followersusers</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only. The response includes the count of users that are following you.

Example

Request

```
HEAD http://webexsocialserver/api/quad/rest/users/@me/follow/@followersusers
```

Response

```
HTTP/1.1 204 No Content
x-quad-count: 2
```

Get Information about Users that are Following the Designated User

Purpose Retrieves Follow resources that have as the destination the users that are following the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/follow/@followersusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve Follow resources that have as the destination the users that are following this user. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for detailed information about using these parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response ([Table 3-24 on page 3-24](#) describes these fields).
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource.
 - The **sortOrder** query parameter is supported for sorting resources in a response when the **filterBy** query parameter is not used. In this case, resources can be sorted by the **firstName** or **lastName** field in the User resource, or by the special **timeStamp** field, which designates when the follow relationship was created.

The **sortOrder** query parameter is not supported for sorting resources in a response when the **filterBy** query parameter is used. In this case, resources are sorted in ascending order by the **fullName** field in the User resource.
 - The **startIndex** query parameter is supported for paginating return results.
 - The **count** query parameter is supported for designating the maximum number of items in the return results.
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata.

Request Payload A request payload is not required.

Response

Standard response (see the [“API Responses”](#) section on page 1-15) with a response payload that can include one or more Follow resources (the [“Follow Relationship Resource”](#) section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/follow/@followersusers
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-198.

JSON Example**Request**

```
GET http://quadserverapi/quad/rest/users/1234/follow/@followersusers
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-206.

Get the Number of Users that are Following the Designated User

Purpose Retrieves the number of users that are following the designated user.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
HEAD	<code>http://server[:port_number]/api/quad/rest/users/user_id/follow/@followersusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve the number of users that are following him or her. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities”](#) section on page 1-11.
- *query_parameters*—Query parameters. See the [“Query Parameters”](#) section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter does not apply
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses”](#) section on page 1-15) with an HTTP header only. The response includes the count of users that are following the designated user.

Example

Request

```
HEAD http://webexsocialserver/api/quad/rest/users/1234/follow/@followersusers
```

Response

```
HTTP/1.1 204 No Content
x-quad-count: 2
```

Get Information about Users that You are Following

Purpose Retrieves Follow resources that have as the destination the users that you are following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/follow/@followingusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for detailed information about using these parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response (Table 3-24 on page 3-24 describes these fields).
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource.
 - The **sortOrder** query parameter is supported for sorting resources in a response when the **filterBy** query parameter is not used. In this case, resources can be sorted by the **firstName** or **lastName** field in the User resource, or by the special **timeStamp** field, which designates when the follow relationship was created.
 The **sortOrder** query parameter is not supported for sorting resources in a response when the **filterBy** query parameter is used. In this case, resources are sorted in ascending order by the **fullName** field in the User resource.
 - The **startIndex** query parameter is supported for paginating return results.
 - The **count** query parameter is supported for designating the maximum number of items in the return results.
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata.

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one or more Follow resources (the “[Follow Relationship Resource](#)” section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/follow/@followingusers
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-198.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/follow/@followingusers
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-206.

Get the Number of Users that You are Following

Purpose Retrieves the number of users that you are following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
HEAD	<code>http://server[:port_number]/api/quad/rest/users/@me/follow/@followingusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query does not apply
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the “[API Responses](#)” section on page 1-15) with an HTTP header only. The response includes the count of users that you are following.

Example

Request

```
HEAD http://webexsocialserver/api/quad/rest/users/@me/follow/@followingusers
```

Response

```
HTTP/1.1 204 No Content
x-quad-count: 2
```

Get Information about Users that the Designated User is Following

Purpose Retrieves Follow resources that have as the destination the users that the designated user is following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/user_id/follow/@followingusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve Follow resources that have as the destination the users this user is following. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities”](#) section on page 1-11.
- *query_parameters*—Query parameters. See the [“Query Parameters”](#) section on page 1-20 for detailed information about using these parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response ([Table 3-24 on page 3-24](#) describes these fields).
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource.
 - The **sortOrder** query parameter is supported for sorting resources in a response when the **filterBy** query parameter is not used. In this case, resources can be sorted by the **firstName** or **lastName** field in the User resource, or by the special **timeStamp** field, which designates when the follow relationship was created.

The **sortOrder** query parameter is not supported for sorting resources in a response when the **filterBy** query parameter is used. In this case, resources are sorted in ascending order by the **fullName** field in the User resource.

- The **startIndex** query parameter is supported for paginating return results.
- The **count** query parameter is supported for designating the maximum number of items in the return results.
- The **meta** query parameter is supported with the **follow** value for requesting relationship metadata.

Request Payload A request payload is not required.

Response

Standard response (see the [“API Responses”](#) section on page 1-15) with a response payload that can include one or more Follow resources (the [“Follow Relationship Resource”](#) section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/follow/@followingusers
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-198.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/1234/follow/@followingusers
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-206.

Get the Number of Users that the Designated User is Following

Purpose Retrieves the number of users that the designated user is following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
HEAD	<code>http://server[:port_number]/api/quad/rest/users/user_id/follow/@followingusers[?query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom to retrieve the number of users that he or she is following. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).
- *query_parameters*—Query parameters. See the [“Query Parameters” section on page 1-20](#) for information about using query parameters. For this operation:
 - The **fields** query parameter does not apply
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only. The response includes the count of users that the designated user is following.

Example

Request

```
HEAD http://webexsocialserver/api/quad/rest/users/1234/follow/@followingusers
```

Response

```
HTTP/1.1 204 No Content
x-quad-count: 2
```

Remove Your Following of Another Cisco WebEx Social User

Purpose Remove the designated follow relationship in which you are following another Cisco WebEx Social user. After you execute this operation, you no longer follow the user, but the user may still be following you.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
DELETE	<code>http://server[:port_number]/api/quad/rest/users/@me/follow/follow_id</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *follow_id*—Identifier of the Follow resource that defines the relationship to remove. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload

A request payload is not required.

Response

Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only.

Example

Request

```
DELETE http://webexsocialserver/api/quad/rest/users/@me/follow/user:1234,user:9876
```

Response

```
HTTP/1.1 204 No Content
```


Get Information about Users that You and the Designated User Both are Following

Purpose Retrieves Follow resources that have as the destination the users you and a designated user both are following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/users/@me/mutualfollow/@followingusers?mutualUserId=user_id[&query_parameters]</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom you want to retrieve users that you both are following. For information about obtaining the identifier that you need, see the “[Identifiers of Cisco WebEx Social Entities](#)” section on page 1-11.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for detailed information about using these parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Follow resource to include in a response ([Table 3-24 on page 3-24](#) describes these fields).
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource.
 - The **sortOrder** query parameter is supported for sorting resources in a response when the **filterBy** query parameter is not used. In this case, resources can be sorted by the **firstName** or **lastName** field in the User resource, or by the special **timeStamp** field, which designates when the follow relationship was created.

The **sortOrder** query parameter is not supported for sorting resources in a response when the **filterBy** query parameter is used. In this case, resources are sorted in ascending order by the **fullName** field in the User resource.
 - The **startIndex** query parameter is supported for paginating return results.
 - The **count** query parameter is supported for designating the maximum number of items in the return results.
 - The **meta** query parameter is supported with the **follow** value for requesting relationship metadata.

Request Payload A request payload is not required.

Response

Standard response (see the [“API Responses”](#) section on page 1-15) with a response payload that can include one or more Follow resources (the [“Follow Relationship Resource”](#) section on page 3-23 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/mutualfollow/
    @followingusers?mutualUserId=1234
Accept:application/xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: XML Format, Multiple Resources Returned”](#) section on page 16-198.

JSON Example**Request**

```
GET http://webexsocialserver/api/quad/rest/users/@me/mutualfollow/
    @followingusers?mutualUserId=1234
Accept:application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

The response can also include the Follow resource. For an example, see the [“Follow Resource Response: JSON Format, Multiple Resources Returned”](#) section on page 16-206.

Get the Number of Users that You and the Designated User Both are Following

Purpose Retrieves the number of users that you and a designated user both are following.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
HEAD	<code>http://server[:port_number]/api/quad/rest/users/@me/mutualfollow/@followingusers?mutualUserId=user_id</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *user_id*—Identifier of the user for whom you want to retrieve the number of users that you both are following. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only. The response includes the count of users that are being followed both by you and the designated user.

Example

Request

```
HEAD http://webexsocialserver/api/quad/rest/users/@me/mutualfollow/
@followingusers?mutualUserId=1234
```

Response

```
HTTP/1.1 204 No Content
x-quad-count: 2
```

Create a Like Relationship for Yourself

Purpose Creates a like relationship in which you like a Cisco WebEx Social post.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
POST	POST <code>http://server[:port_number]/api/quad/rest/users/@me/like</code> [?query_parameters]

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *query_parameters*—Query parameters. See the “[Query Parameters](#)” section on page 1-20 for information about using query parameters. For this operation:
 - The **fields** query parameter is supported for selecting fields from the Like resource to include in a response ([Table 3-26 on page 3-25](#) describes these fields)
 - The **filterBy** query parameter is supported for filtering resources in a response by the **fullName** field in the User resource
 - The **sortOrder** query parameter does not apply
 - The **startIndex** query parameter does not apply
 - The **count** query parameter does not apply
 - The **meta** query parameter does not apply

Request Payload

The request payload includes fields from the Like resource ([Table 3-26 on page 3-25](#) describes these fields):

Required Fields	Optional Fields
destination ¹	—

1. The **destination** field defines a resource that represents the post that you like. The **id** and the **resource** fields in this resource are required. Other fields are not used.

Response

Standard response (see the “[API Responses](#)” section on page 1-15) with a response payload that can include one Like resource (the “[Like Relationship Resource](#)” section on page 3-25 describes this resource). XML and JSON formats are supported.

XML Example**Request**

```
POST http://quadserver//api/quad/rest/users/@me/like
Accept:application/xml
Content-Type:application/xml
```

```
<like>
  <destination>
    <resource>http://quad.cisco.com/schema/1.0/post</resource>
    <id>123</id>
  </destination>
</like>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://webexsocialserver/api/quad/rest/users/@me/like/user:1234,post:600007
```

The response can also include the Like resource. For an example, see the [Like Resource Response with Post Resource in Destination Field: XML Format, One Resource Returned, page 16-212](#).

JSON Example**Request**

```
POST http://quadserver//api/quad/rest/users/@me/like
Accept:application/json
Content-Type:application/json
```

```
{
  "destination": {
    "id": "123",
    "resource": "http://quad.cisco.com/schema/1.0/post"
  }
}
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://webexsocialserver/api/quad/rest/users/@me/like/user:1234,post:600007
```

The response can also include the Like resource. For an example, see the [Like Resource Response with Post Resource in Destination Field: JSON Format, One Resource Returned, page 16-223](#).

Remove a Like Relationship for Yourself

Purpose Removes the designated like relationship in which you like a Cisco WebEx Social post.

Availability Cisco WebEx Social 3.0 and later.

Request

HTTP Method	URI
DELETE	<code>http://server[:port_number]/api/quad/rest/users/@me/like/like_id</code>

where:

- *server*—Host name or IP address of the Cisco WebEx Social server.
- *port_number*—Port number on which the Cisco WebEx Social server listens for API requests. The default value is **80**.
- *like_id*—Identifier of the Like resource that defines the relationship to remove. For information about obtaining the identifier that you need, see the [“Identifiers of Cisco WebEx Social Entities” section on page 1-11](#).

Request Payload A request payload is not required.

Response Standard response (see the [“API Responses” section on page 1-15](#)) with an HTTP header only.

Example

Request

```
DELETE http://webexsocialserver/api/quad/rest/users/@me/like/user:1234,post:100001
```

Response

```
HTTP/1.1 204 No Content
```