



## CHAPTER 6

# Membership Request Resource Operations

This chapter describes the Cisco WebEx Social API Membership Request resource operations. These operations let you retrieve and update information about a user request for membership in a Cisco WebEx Social community.

[Table 6-1](#) provides a summary of the Membership Request 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

**Table 6-1** *Membership Request Resource Operations Summary*

Operation and Reference	Purpose
<a href="#">Get Information about a Membership Request, page 6-2</a>	Retrieves information about the designated membership request
<a href="#">Update a Membership Request, page 6-5</a>	Updates the designated membership request

# Get Information about a Membership Request

**Purpose** Retrieves information about the designated membership request.

**Availability** Cisco WebEx Social 3.0 and later.

## Request

HTTP Method	URI
GET	<code>http://server[:port_number]/api/quad/rest/membershiprequests/membership_request_id</code>

where:

<i>server</i>	Host name or IP address of the Cisco WebEx Social server.
<i>port_number</i>	Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.
<i>membership_request_id</i>	Identifier of the membership request for which to retrieve information. For information about obtaining the identifier that you need, see the <a href="#">“Identifiers of Cisco WebEx Social Entities”</a> section on page 1-10.

## Request Payload

A request payload is not required.

## Response

Standard response (see the [“API Responses”](#) section on page 1-14) with a response payload that can include one MembershipRequest resource (the [“MembershipRequest Resource”](#) section on page 3-20 describes this resource). XML and JSON formats are supported.

## XML Example

### Request

```
GET http://webexsocialserver/api/quad/rest/membershiprequests/210747
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>1</itemsPerPage>
<totalResults>0</totalResults>
<filtered>>false</filtered>
<sorted>>false</sorted>
<entry>
  <membershiprequest>
    <approver xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
```

```

<community>
  <description>second</description>
  <id>4710003</id>
  <name>Dos</name>
  <shortName>/dos</shortName>
  <state>live</state>
  <type>restricted</type>
</community>
<createDate>1319730564972</createDate>
<id>110499</id>
<replyComments></replyComments>
<replyDate>0</replyDate>
<requestComments></requestComments>
<requester>
  <firstName>Pat</firstName>
  <fullName>Pat Kim</fullName>
  <id>1234</id>
  <jobTitle></jobTitle>
  <lastName>Kim</lastName>
  <screenName>patkim</screenName>
</requester>
<resource>http://quad.cisco.com/schema/1.0/membershiprequest</resource>
<status>pending</status>
</membershiprequest>
</entry>
</ns2:response>

```

**JSON Example****Request**

```

GET http://webexsocialserver/api/quad/rest/membershiprequests/210747
Accept:application/json

```

**Response**

```

HTTP/1.1 200 OK
Content-Type: application/json
{
  "entry": [
    {
      "data": {
        "approver": null,
        "community": {
          "description": "second",
          "id": 4710003,
          "name": "Dos",
          "shortName": "/dos",
          "state": "live",
          "type": "restricted"
        },
        "createDate": 1319730564972,
        "id": 110499,
        "replyComments": "",
        "replyDate": 0,
        "requestComments": "",
        "requester": {
          "firstName": "Pat",
          "fullName": "Pat Kim",
          "id": 1234,
          "images": [ ],
          "jobTitle": "",
          "lastName": "Kim",
          "screenName": "patkim"
        }
      }
    }
  ]
}

```

```
    },
    "resource": "http://quad.cisco.com/schema/1.0/membershiprequest",
    "status": "pending"
  }
}
],
"filtered": false,
"itemsPerPage": 1,
"serverMessages": null,
"sorted": false,
"startIndex": 0,
"statusCode": 200,
"totalResults": 0
}
```

# Update a Membership Request

**Purpose** Updates the designated membership request.

**Availability** Cisco WebEx Social 3.0 and later.

## Request

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

where:

<i>server</i>	Host name or IP address of the Cisco WebEx Social server.
<i>port_number</i>	Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.
<i>membership_request_id</i>	Identifier of the membership request. For information about obtaining the identifier that you need, see the <a href="#">“Identifiers of Cisco WebEx Social Entities” section on page 1-10.</a>

## Request Payload

The request payload includes fields from the Membership Request resource ([Table 3-22 on page 3-20](#) describes these fields):

Required Fields	Optional Fields
status	replyComments

## Response

Standard response (see the [“API Responses” section on page 1-14](#)) with an HTTP header only.

## XML Example

### Request

```
PUT http://webexsocialserver/api/quad/rest/membershiprequests/210747
Accept: application/xml
Content-Type: application/xml
```

```
<membershiprequest>
  <status>approved</status>
</membershiprequest>
```

### Response

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

**JSON Example****Request**

```
PUT http://webexsocialserver/api/quad/rest/membershiprequests/210747
Accept:application/json
Content-Type: application/json
```

```
{
  "status": "approved"
}
```

**Response**

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