



CHAPTER 15

Search Operation

This chapter describes the Cisco WebEx Social API Search operation. This operation lets you search for content, users, or communities through all Cisco WebEx Social elements, including information (posts, blogs, comments, attachments, and so on), users, and communities. This operation returns results that comply with the OpenSearch specification.

This operation is available in Cisco WebEx Social 3.0 and later.

By default, the Search operation performs as follows, but you can adjust this performance by using options and query parameters with the command:

- Searches all Cisco WebEx Social elements
- Does not search for specific text strings
- Does not filter results
- Presents search results in order from most relevant to least relevant
- Does not include metadata in search results
- Includes 25 items
- Searches from the beginning of January 1, 1970 through the current date and time

The Search operation returns results in JSON or Atom formats.

This chapter includes these topics:

- [Using the Search Operation, page 15-1](#)
- [Search Operation Response, page 15-11](#)
- [Using Query Tags to Refine a Search, page 15-11](#)
- [Obtaining the OpenSearchDescription document, page 15-12](#)
- [Examples, page 15-12](#)

Using the Search Operation

The following sections describe how to use the Search operation:

- [Request, page 15-2](#)
- [Query Parameters, page 15-2](#)

Request

To execute the Search operation, submit an HTTP request in the following format:

```
GET http://server[:port_number]/api/quad/rest/search/[@type][?q=criteria]
[{{?!&}filterBy={information|users|communities|fileTypes|scope}&filterValue=value}...]
[{{?!&}sortBy={relevancelmodified}}][{{?!&}searchMetadata={true|false}}]
[{{?!&}startIndex=position][{{?!&}&count=total_items][{{?!&}startTime=start&endTime=end]
```

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**.
- *type*—Limits the results to a Cisco WebEx Social element type. Valid values are:
 - **information**—Causes the operation to return only results that come from posts, blogs, comments, attachments, and so on
 - **users**—Causes the operation to return only results that are users
 - **communities**—Causes the operation to return only results that are communities

If you do not specify an element type, this operation can return results from all Cisco WebEx Social element types.

- *criteria, value, position, total_items, start, end*—Query parameter arguments. For detailed explanations, see the “[Query Parameters](#)” section on page 15-2.

Query Parameters

You can use query parameters with the Search operation to designate criteria that data in a response must match, and to sort and paginate data that is returned.

The first query parameter that you include in a Search request must be preceded by a question mark (?). When you include multiple query parameters in request, separate each parameter with an ampersand (&). Query parameters are case-sensitive.

The following section describe the query parameters for the Search operation:

- [q, page 15-2](#)
- [filterBy, page 15-8](#)
- [sortBy, page 15-9](#)
- [searchMetadata, page 15-9](#)
- [startIndex, page 15-10](#)
- [count, page 15-10](#)
- [startTime, page 15-10](#)

q

The **q** query parameter lets you designate one or more string to search for. In general, the search criteria that you designate with this parameter use the Lucene query syntax.

Enter this parameter in the following format:

q=*criteria*

Table 15-1 describe the search criteria that you can use.

Table 15-2 on page 15-7 describes techniques that you can use for additional control over search criteria.

To include any of the following special characters in a search, the characters must be URL encoded:

+ - & ! ! () { } [] ^ " ~ * ? : \

Table 15-1 Search Criteria

Description	Valid <i>criteria</i>	Examples
Single word	Enter any group of characters without spaces. This string is not case sensitive.	q=Cisco Returns results that include Cisco (in any case).
Exact phrase	Enclose the phrase within double quotation marks (" "). This phrase is case sensitive.	q="Cisco WebEx Social" Returns results that include Cisco WebEx Social .
Wildcard—single character	Use a question mark (?) anywhere except as the first character in a single word or string to represent any character.	q=fla? Returns results that include flag, flan, flat , and so on.
Wildcard—multiple characters	Use an asterisk (*) anywhere except as the first character in a single word or string to represent any sequence of characters.	q=part* Returns results that include partake, partner, party , and so on.
Multiple terms—all	Use AND between any number of single words an exact phrases in any combination.	q=Cisco AND WebEx Returns results that include both Cisco and WebEx (in any case).
Multiple terms—any	Use OR between any number of single words an exact phrases in any combination.	q=Cisco OR WebEx Returns results that include Cisco, WebEx , or both terms (in any case).
Required term	Use a plus sign (+).	q=+Cisco WebEx Returns results that include Cisco and that can include WebEx (in any case). q=+Cisco +WebEx Returns results that include both Cisco and WebEx (in any case).
Unwanted term	Use NOT or a dash (-) in front of each term that should not be included.	q=WebEx NOT open q=WebEx -open Returns results that include WebEx (in any case) and do not include open (in any case).

Table 15-1 Search Criteria (continued)

Description	Valid criteria	Examples
Proximity search	Enclose a pair of single words within double quotation marks (" "), and follow this string with a tilde (~) and a numeric value of 1 or greater. The system returns the designated terms if they are separated by no more than the number of words that the numeric value defines.	q="WebEx openapi"~10 Returns results in which WebEx (in any case) and openapi (in any case) are within 10 words of each other.
Fuzzy search	Use a tilde (~) at the end of a single word. Optionally follow the tilde with a numeric value from 0 to 1 . The system uses the Levenshtein Distance algorithm to return words or terms that are similar in spelling to the string that you designate. The numeric value controls how similar the items must be. A lower value indicates less similar, and a higher number indicates more similar. If you do not enter a number, the system assumes a value of 0.5 .	q=roam~ Could return results that include road , roar , and so on.
Application type search	Use portletType: followed by the one of the following types to limit the search to applications of the specified type: <ul style="list-style-type: none"> • Blogs • Bookmarks • Calendar • Comment • Community • Document Library • Image Gallery • Media Gallery • Message Boards • People • Post • Post Attachment • Web Contents • Wiki 	q=portletType:Calendar Returns content from calendar applications only.

Table 15-1 Search Criteria (continued)

Description	Valid criteria	Examples
Content search	Use content: followed by the content of an element. Can be a single word or an exact phrase.	q=content:WebEx Returns content that includes WebEx (in any case). q=content:"Cisco WebEx" Returns content that includes Cisco WebEx .
Created date search	Use createdDate: followed by the date that an element was created, in yyyy[mmdd] format.	q=createdDate:20121015 Returns content that was created on October 15, 2012. q=createdDate:2012 Returns content that was created in the year 2012.
E- mail address search	Use emailAddress: followed by the e-mail address of a user.	q=emailAddress:kim@here.com Returns the user with the e-mail address kim@here.com .
Expertise search	Use expertise: followed by the expertise of a user. Can be either a single word or an exact phrase.	q=expertise:blogging Returns users with an expertise of blogging . q=expertise:"News blogging" Returns users with an expertise of News blogging .
File type search	Use fileType: followed by a file extension, such as txt or doc, that designates a file type.	q=fileType:txt Returns files with the extension txt .
Job title search	Use jobTitle: followed by the job title of a user. Can be either a single word or an exact phrase.	q=jobTitle:manager Returns users with the job title manager . q=jobTitle:"Senior manager" Returns users with the job title Senior manager .
Modified date search	Use modifiedDate: followed by the date that an element was last modified, in yyyy[mmdd] format.	q=modifiedDate:20120710 Returns content that was modified on July 7, 2012. q=modifiedDate:20120710 Returns content that was modified in the year 2012.
Screen name search	Use screenName: followed by the Cisco WebEx Social screen name of a user. The screen name is not case sensitive.	q=screenName:patkim Returns users with the screen name patkim .

Table 15-1 Search Criteria (continued)

Description	Valid criteria	Examples
Tag search	Single tag: Use tag: followed by the desired tag. Tag values are not case sensitive.	q=tag:openapi Returns elements that include the tag openapi (in any case).
	Multiple tag search: Multiple tag, any: Use tag: to identify each tag and use OR between each tag, or use tag: to identify each tag and separate each tag with a space. Tag values are not case sensitive.	q=tag:cisco OR tag:WebEx q=tag:cisco tag:WebEx Returns elements that include the tag Cisco (in any case), or the tag WebEx (in any case), or both tags.
	Multiple tag, all: Use tag: to identify each tag and use AND between each tag, or use +tag: to identify each tag. Tag values are not case sensitive.	q=tag:Cisco AND tag:WebEx q=+tag:Cisco +tag:WebEx Returns elements that include the tags Cisco and WebEx (in any case).
Title search	Use title: followed by the title of a Cisco WebEx Social element. This title is not case sensitive.	q=title:"Cisco Collaboration" Returns elements with a title that includes the phrase Cisco Collaboration (in any case).
User Id search	Use userId: followed by the Cisco WebEx Social identifier of a user.	q=userId:1357 Returns the user with the identifier 1357 .
User name search	Use userName: followed by the Cisco WebEx Social name of a user who created an element. To use a full name, enclose the name within double quotation marks (" ").	q=userName:Pat Returns users with the name Pat. q=userName:"Pat Kim" Returns users with the name Pat Kim.

You can use a variety of techniques alone or in combination to provide additional control over search criteria. [Table 15-2](#) describes these techniques.

Table 15-2 Search Techniques

Technique	Description	Examples
Clause grouping	Use parentheses (()) to group search clauses to form sub-queries.	<p>q=(WebEx OR collaborate) AND cisco</p> <p>Returns elements that include WebEx and cisco, or collaborate and cisco in any order and in any case.</p> <p>q=(title:WebEx OR title:collaborate) AND title:cisco</p> <p>Returns elements with a title that includes WebEx and cisco, or collaborate and cisco in any order and in any case.</p>
Field grouping	Use parentheses (()) to group multiple search clauses into a single search field.	<p>q=title:(+cisco +"WebEx user")</p> <p>Returns elements with a title that includes both the word cisco in any case and the phrase WebEx user.</p>
Boosting	<p>Boosting a search item makes that item more relevant than other items in the search. To boost an item, use a caret (^) at the end of the item.</p> <p>You can optionally follow the caret with a positive numeric value up to 10, which designates the boost factor for the item. A higher boost value indicates a higher boost factor, which increases the relevance of the item. If you do not specify a value, the system uses a boost factor of 1.</p>	<p>q=cisco WebEx^8</p> <p>Returns elements that include the terms cisco or WebEx (in any case), and makes WebEx the more relevant term.</p> <p>q="my WebEx"^10 cisco</p> <p>Returns elements that include the term my WebEx (in this case) or the word cisco (in any case), and makes my WebEx the more relevant term.</p>
Range searching	<p>Ranges let you find elements that have values within the lower and upper bounds, inclusive, of dates or alphanumeric characters.</p> <p>To search for all elements that were created or modified within a date range, specify the range within brackets ([]) using the format [yyyymmdd TO yyyymmdd].</p> <p>To search for items that are within a range of characters, specify the range within brackets using the format [start_range TO end_range].</p>	<p>q=createdDate:[20110101 TO 20120710]</p> <p>Returns elements that were created from January 1, 2011 through July 10, 2012.</p> <p>q=username:[Kim TO Sam]</p> <p>Returns Cisco WebEx Social user names range from Kim to Sam.</p>

filterBy

The **filterBy** query parameter causes the Search operation to return only results that are filtered by a designated item and value.

If you do not specify this query parameter, results are not filtered.

Enter this query parameter in the following format:

filterBy={**information**|**users**|**communities**|**fileTypes**|**scope**}&**filterValue**=*value*

You can specify multiple filterBy query parameters with the Search operation, but each instance of this query parameter must contain a different filterBy argument. For example:

- Valid—**filterBy=information&filterValue=post&filterBy=user&filterValue=1234**
- Not valid—**filterBy=information&filterValue=post&filterBy=information&filterValue=blog**

Table 15-3 describes the arguments and valid *value* strings for this query parameter.

Table 15-3 Arguments for filterBy Query Parameter

Argument	Description	Valid <i>value</i> Strings
information	Designates that the Search operation returns information results that are filtered as defined by the <i>value</i> string. Information includes posts, blogs, comments, attachments, and so on.	<ul style="list-style-type: none"> • blog • bookmark • calendar • comment • documentLibrary • imageGallery • mediaGallery • messageBoard • micropost • post • postAttachment • socialActivity • video • webContent • wiki
users	Designates that the Search operation returns results that relate to the Cisco WebEx Social user whom the <i>value</i> string defines.	Identifier of the user for whom to return results. For information about obtaining the identifier that you need, see the “Identifiers of Cisco WebEx Social Entities” section on page 1-11.
communities	Designates that the Search operation returns results that relate to the Cisco WebEx Social community that the <i>value</i> string defines.	Identifier of the community for which to return results. For information about obtaining the identifier that you need, see the “Identifiers of Cisco WebEx Social Entities” section on page 1-11.

Table 15-3 Arguments for filterBy Query Parameter (continued)

Argument	Description	Valid <i>value</i> Strings
fileTypes	Designates that the Search operation returns content that contains a file of the type that the <i>value</i> string defines, or returns files of the type that the <i>value</i> string defines.	File extension, such as doc , pdf , or txt .
scope	Designates that the Search operation returns results only from communities to which you belong.	@myCommunities

sortBy

The **sortBy** query parameter determines the order in which the Search operation presents search results. If you do not specify this query parameter, results presented in order of relevance from highest to lowest.

Enter this query parameter in the following format:

sortBy={relevancelmodified}

Table 15-4 describes the arguments for this query parameter.

Table 15-4 Arguments for sortOrder Query Parameter

Argument	Description
relevance	Sorts search results in order from most relevant to least relevant
modified	Sorts search results in chronological order beginning with the most recent modified date and time

searchMetadata

The **searchMetadata** query parameter designates whether the following metadata is included with the results that the Search operation returns:

- **totalResults**—Number of information, community, and user items that the Search operation identifies in Cisco WebEx Social.
- **totalInfo**—Number of information items that the Search operation identifies in Cisco WebEx Social. information items include posts, blogs, comments, attachments, and so on.
- **totalComm**—Number of communities that the Search operation identifies in Cisco WebEx Social.
- **totalUser**—Number of users that the Search operation identifies in Cisco WebEx Social.
- **itemsPerPage**—Number of information, community, and user items that are included in the response.
- **numInfo**—Number of information items that are included in the response.
- **numComm**—Number of communities that are included in the response.
- **numUser**—Number of users that are included in the response.
- **startIndex**—The **startIndex** value, if included in the operation request.

If you do not specify this query parameter, metadata is not included with the return results.

Enter this query parameter in the following format:

searchMetadata={true|false}

Table 15-5 describes the arguments for this query parameter.

Table 15-5 Arguments for searchMetadata Query Parameter

Argument	Description
true	Include metadata with search results
false	Do not include metadata with search results

startIndex

The **startIndex** query parameter designates the position in a list of items of the first item that the Search operation returns. If you do not specify this query parameter, the index starts with the first item in the list.

Enter this query parameter in the following format, where *position* is a number that identifies the position in a list of items of the first item to include in the return results. The number 0 indicates the first item in a list, 1 indicates the second item, and so on:

startIndex=position

count

The **count** query parameter designates the number of consecutive items to include in the return results. If you do not specify this query parameter, 25 consecutive items are returned.

Enter this query parameter in the following format, where count is the number of consecutive items to return:

count=total_items

startTime

The **startTime** query parameter lets you designate a period of consecutive days within which an element must have been created or modified to be included in the return results. If you do not specify this parameter, elements that were created on any day from January 1, 1970 through the day on which you execute the Search operation are returned.

Enter this query parameter in the following format:

startTime=start&endTime=end

Table 15-6 describes the arguments for this query parameter.

Table 15-6 Arguments for startTime Query Parameter

Argument	Description
<i>start</i>	First day of the period of consecutive days, in UTC format. For example 1332720000000 indicates March 26, 2012, 00:00:00 GMT.
<i>end</i>	Last day of the period of consecutive days, in UTC format. For example 1332806400000 indicates March 27, 2012, 00:00:00 GMT.

Search Operation Response

The Search operation returns a response that includes the results of the requested search. Results are presented in descending order according to relevance and modification date and time. JSON and Atom formats are supported.

A response also includes one or more query tags, which provide information that you can use to refine a search. For more information, see the [“Using Query Tags to Refine a Search”](#) section on page 15-11.

Using Query Tags to Refine a Search

A response to a Search operation request includes one or more query tags.

[Table 15-7](#) describes the items that a query tag contains. [Example 15-1](#) shows a query tag in JSON format. [Example 15-2](#) shows a query tag in Atom format.

Table 15-7 Items Query Tag

Item	Description
count	The number of results that will be returned if you use the suggested query
filterBy	Value that you can use with the filterBy query parameter to further refine a search
filterDisplayName	String value that identifies the name of the corresponding item that is returned
filterValue	Value that you can use with the filterValue query parameter to further refine a search
searchMetadata	Value of the searchMetadata query parameter (the default value is false)
searchTerm	Value of the q query parameter, if any, that was used in the search

Example 15-1 Query Tag in JSON Format

```
"os:Query": [
  {
    "cisco:filterValue": "post",
    "cisco:filterDisplayName": "post",
    "searchTerms": "asdf",
    "count": "3",
    "role": "subset",
    "cisco:filterBy": "information",
    "cisco:searchMetadata": "true"
  }
]
```

Example 15-2 Query Tag in Atom Format

```
<os:Query cisco:filterBy="information" cisco:filterValue="post"
cisco:filterDisplayName="post" count="3" cisco:searchMetadata="true" role="subset"
searchTerms="asdf" />
```

You can use the information that a query tag provides to refine a search. For example:

1. Execute a general Search request, such as:

GET <http://webexsocialserver/api/quad/rest/search?q=openapi>

This request might receive a response that includes the following query tag:

```
<os:Query cisco:filterBy="information" cisco:filterValue="post"
cisco:filterDisplayName="post" count="3" cisco:searchMetadata="true" role="subset"
searchTerms="openapi" />
```

2. Use the information in the query tag to execute a refined search:

GET <http://webexsocialserver/api/quad/rest/search?q=openapi&filterBy=information&filterValue=post&searchMetadata=true>

Obtaining the OpenSearchDescription document

The Cisco WebEx OpenSearchDescription document is an XML document that provides a description of the search engine for the Search operation.

To obtain this document, execute the Search operation with the header **Accept:application/opensearchdescription+xml**.

For example:

```
GET http://webexsocialserver/api/quad/rest/search
Accept:application/opensearchdescription+xml
```

Examples

The following sections show examples of using the Search operation.

- [JSON Examples, page 15-12](#)
- [Atom Examples, page 15-18](#)

JSON Examples

Example 1 Request

```
GET http://webexsocialserver/api/quad/rest/search?q=asdf&searchMetadata=true&count=2
Accept:application/json
```

Example 1 Response

```
{
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "cisco:numComm": "0",
    "author": {
      "name": "http://quad.cisco.com"
    },
    "os:itemsPerPage": "2",
    "title": {
      "content": "http://quad.cisco.com OpenSearch",
      "type": "text"
    },
    "os:startIndex": "0",
    "updated": "2012-06-28T16:51:31.435Z",
    "entry": [
      {
        "summary": {
          "content": "<em>asdf</em>",
          "type": "text"
        }
      }
    ]
  }
}
```

```

    },
    "id": "8100001",
    "author": {
      "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
      "cisco:id": "10467",
      "uri": "/users"
    },
    "category": {
      "term": "information"
    },
    "title": {
      "content": "asdfasdf",
      "type": "text"
    },
    "cisco:resource": "http://quad.cisco.com/schema/1.0/post",
    "updated": "2012-06-12T13:41:51.000Z",
    "cisco:score": "1.0",
    "link": {
      "href": "/posts/8100001"
    },
    "published": "2012-06-12T13:41:51.000Z"
  },
  {
    "summary": {
      "content": "<em>asdf</em>",
      "type": "text"
    },
    "id": "8100003",
    "author": {
      "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
      "cisco:id": "10467",
      "uri": "/users"
    },
    "category": {
      "term": "information"
    },
    "title": {
      "content": "test post",
      "type": "text"
    },
    "cisco:resource": "http://quad.cisco.com/schema/1.0/post",
    "updated": "2012-06-13T12:01:40.000Z",
    "cisco:score": "0.875",
    "link": {
      "href": "/posts/8100003"
    },
    "published": "2012-06-13T11:59:37.000Z"
  }
],
"xmlns:cisco": "http://quad.cisco.com",
"cisco:numInfo": "2",
"os:totalResults": "3",
"os:Query": [
  {
    "cisco:filterValue": "post",
    "cisco:filterDisplayName": "post",
    "searchTerms": "asdf",
    "count": "3",
    "role": "subset",
    "cisco:filterBy": "information",
    "cisco:searchMetadata": "true"
  }
],
{
  "cisco:filterValue": "8510031",

```

```

    "cisco:filterDisplayName": "test pat open community",
    "searchTerms": "asdf",
    "count": "1",
    "role": "subset",
    "cisco:filterBy": "communities",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "10467",
    "cisco:filterDisplayName": "Pat Kim",
    "searchTerms": "asdf",
    "count": "3",
    "role": "subset",
    "cisco:filterBy": "users",
    "cisco:searchMetadata": "true"
  }
],
"cisco:numUser": "0",
"cisco:totalComm": "0",
"cisco:totalUser": "0",
"xmlns:os": "http://a9.com/-/spec/opensearch/1.1/",
"cisco:totalInfo": "3"
}
}

```

Example 2 Request

```

GET http://webexsocialserver/api/quad/rest/search?q=pat&searchMetadata=true&count=2
Accept:application/json

```

Example 2 Response

```

{
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "cisco:numComm": "0",
    "author": {
      "name": "http://quad.cisco.com"
    },
    "os:itemsPerPage": "2",
    "title": {
      "content": "http://quad.cisco.com OpenSearch",
      "type": "text"
    },
    "os:startIndex": "0",
    "updated": "2012-06-28T16:54:07.389Z",
    "entry": [
      {
        "summary": {
          "content": "<em>Pat</em> Kim &lt;pat@quad.cisco.com&gt;",
          "type": "text"
        },
        "id": "10467",
        "author": {
          "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
          "email": "pat@quad.cisco.com",
          "cisco:id": "10467",
          "uri": "/users"
        },
        "category": {
          "term": "users"
        },
        "title": {
          "content": "<em>Pat</em> Kim",

```

```

        "type": "text"
    },
    "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
    "updated": "2012-06-28T09:48:46.000Z",
    "cisco:score": "1.0",
    "link": {
        "href": "/users/10467"
    },
    "cisco:tags": "rest"
},
{
    "summary": {
        "type": "text"
    },
    "id": "16110047",
    "author": {
        "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
        "cisco:id": "10467",
        "uri": "/users"
    },
    "category": {
        "term": "information"
    },
    "title": {
        "content": "Orange.gif",
        "type": "text"
    },
    "cisco:resource": "http://quad.cisco.com/schema/1.0/attachment",
    "updated": "2012-06-13T12:01:40.000Z",
    "cisco:score": "0.015599317",
    "link": {
        "href": "/attachments/16110047"
    },
    "published": "2012-06-13T12:01:40.000Z"
}
],
"xmlns:cisco": "http://quad.cisco.com",
"cisco:numInfo": "1",
"os:totalResults": "10",
"os:Query": [
    {
        "cisco:filterValue": "imageGallery",
        "cisco:filterDisplayName": "imageGallery",
        "searchTerms": "pat",
        "count": "2",
        "role": "subset",
        "cisco:filterBy": "information",
        "cisco:searchMetadata": "true"
    },
    {
        "cisco:filterValue": "postAttachment",
        "cisco:filterDisplayName": "postAttachment",
        "searchTerms": "pat",
        "count": "1",
        "role": "subset",
        "cisco:filterBy": "information",
        "cisco:searchMetadata": "true"
    },
    {
        "cisco:filterValue": "post",
        "cisco:filterDisplayName": "post",
        "searchTerms": "pat",
        "count": "4",
        "role": "subset",

```

```

    "cisco:filterBy": "information",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "micropost",
    "cisco:filterDisplayName": "micropost",
    "searchTerms": "pat",
    "count": "2",
    "role": "subset",
    "cisco:filterBy": "information",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "8510031",
    "cisco:filterDisplayName": "test pat open community",
    "searchTerms": "pat",
    "count": "3",
    "role": "subset",
    "cisco:filterBy": "communities",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "10467",
    "cisco:filterDisplayName": "Pat Kim",
    "searchTerms": "pat",
    "count": "10",
    "role": "subset",
    "cisco:filterBy": "users",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "gif",
    "cisco:filterDisplayName": "gif",
    "searchTerms": "pat",
    "count": "2",
    "role": "subset",
    "cisco:filterBy": "fileTypes",
    "cisco:searchMetadata": "true"
  },
  {
    "cisco:filterValue": "jpg",
    "cisco:filterDisplayName": "jpg",
    "searchTerms": "pat",
    "count": "1",
    "role": "subset",
    "cisco:filterBy": "fileTypes",
    "cisco:searchMetadata": "true"
  }
],
"cisco:numUser": "1",
"cisco:totalComm": "0",
"cisco:totalUser": "1",
"xmlns:os": "http://a9.com/-/spec/opensearch/1.1/",
"cisco:totalInfo": "9"
}
}

```

Example 3 Request

```

GET http://webexsocialserver/api/quad/rest/search?q=test&searchMetadata=true&count=2
Accept:application/json

```


Example 3 Response

```

{
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "cisco:numComm": "0",
    "author": {
      "name": "http://quad.cisco.com"
    },
    "os:itemsPerPage": "1",
    "title": {
      "content": "http://quad.cisco.com OpenSearch",
      "type": "text"
    },
    "os:startIndex": "0",
    "updated": "2012-06-28T16:56:08.778Z",
    "entry": {
      "summary": {
        "content": "asdf",
        "type": "text"
      },
      "id": "8100003",
      "author": {
        "cisco:resource": "http://quad.cisco.com/schema/1.0/user",
        "cisco:id": "10467",
        "uri": "/users"
      },
      "category": {
        "term": "information"
      },
      "title": {
        "content": "<em>test</em> post",
        "type": "text"
      },
      "cisco:resource": "http://quad.cisco.com/schema/1.0/post",
      "updated": "2012-06-13T12:01:40.000Z",
      "cisco:score": "1.0",
      "link": {
        "href": "/posts/8100003"
      },
      "published": "2012-06-13T11:59:37.000Z"
    },
    "xmlns:cisco": "http://quad.cisco.com",
    "cisco:numInfo": "1",
    "os:totalResults": "1",
    "os:Query": [
      {
        "cisco:filterValue": "post",
        "cisco:filterDisplayName": "post",
        "searchTerms": "test",
        "count": "1",
        "role": "subset",
        "cisco:filterBy": "information",
        "cisco:searchMetadata": "true"
      },
      {
        "cisco:filterValue": "10467",
        "cisco:filterDisplayName": "Pat Kim",
        "searchTerms": "test",
        "count": "1",
        "role": "subset",
        "cisco:filterBy": "users",
        "cisco:searchMetadata": "true"
      }
    ]
  }
}

```

```

],
  "cisco:numUser": "0",
  "cisco:totalComm": "0",
  "cisco:totalUser": "0",
  "xmlns:os": "http://a9.com/-/spec/opensearch/1.1/",
  "cisco:totalInfo": "1"
}
}
}

```

Atom Examples

Example 1 Request

```

ET http://webexsocialserver/api/quad/rest/search?q=asdf&searchMetadata=true&count=2
Accept:application/atom+xml

```

Example 1 Response

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
xmlns:cisco="http://quad.cisco.com">
  <title type="text">http://quad.cisco.com OpenSearch</title>
  <updated>2012-06-28T16:48:46.572Z</updated>
  <author>
    <name>http://quad.cisco.com</name>
  </author>
  <cisco:totalInfo>3</cisco:totalInfo>
  <cisco:totalComm>0</cisco:totalComm>
  <cisco:totalUser>0</cisco:totalUser>
  <cisco:numInfo>2</cisco:numInfo>
  <cisco:numComm>0</cisco:numComm>
  <cisco:numUser>0</cisco:numUser>
  <os:totalResults>3</os:totalResults>
  <os:itemsPerPage>2</os:itemsPerPage>
  <os:startIndex>0</os:startIndex>
  <os:Query cisco:filterBy="information" cisco:filterValue="post"
cisco:filterDisplayName="post" count="3" cisco:searchMetadata="true" role="subset"
searchTerms="asdf" />
  <os:Query cisco:filterBy="communities" cisco:filterValue="8510031"
cisco:filterDisplayName="test pat open community" count="1" cisco:searchMetadata="true"
role="subset" searchTerms="asdf" />
  <os:Query cisco:filterBy="users" cisco:filterValue="10467" cisco:filterDisplayName="Pat
Kim" count="3" cisco:searchMetadata="true" role="subset" searchTerms="asdf" />
  <entry>
    <id>8100001</id>
    <author>
      <cisco:id>10467</cisco:id>
      <uri>/users</uri>
      <cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
    </author>
    <summary type="text">&lt;em>asdf&lt;/em>
  </summary>
  <updated>2012-06-12T13:41:51.000Z</updated>
  <published>2012-06-12T13:41:51.000Z</published>
  <title type="text">asdfasdf</title>
  <cisco:score>1.0</cisco:score>
  <category term="information" />
  <cisco:resource>http://quad.cisco.com/schema/1.0/post</cisco:resource>
  <link href="/posts/8100001" />
</entry>
<entry>
  <id>8100003</id>
  <author>

```

```

    <cisco:id>10467</cisco:id>
    <uri>/users</uri>
    <cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
  </author>
  <summary type="text">&lt;em>asdf&lt;/em></summary>
</summary>
<updated>2012-06-13T12:01:40.000Z</updated>
<published>2012-06-13T11:59:37.000Z</published>
<title type="text">test post</title>
<cisco:score>0.875</cisco:score>
<category term="information" />
<cisco:resource>http://quad.cisco.com/schema/1.0/post</cisco:resource>
<link href="/posts/8100003" />
</entry>
</feed>

```

Example 2 Request

```

GET http://webexsocialserver/api/quad/rest/search?q=pat&searchMetadata=true&count=2
Accept:application/atom+xml

```

Example 2 Response

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
xmlns:cisco="http://quad.cisco.com">
  <title type="text">http://quad.cisco.com OpenSearch</title>
  <updated>2012-06-28T16:54:47.364Z</updated>
  <author>
    <name>http://quad.cisco.com</name>
  </author>
  <cisco:totalInfo>9</cisco:totalInfo>
  <cisco:totalComm>0</cisco:totalComm>
  <cisco:totalUser>1</cisco:totalUser>
  <cisco:numInfo>1</cisco:numInfo>
  <cisco:numComm>0</cisco:numComm>
  <cisco:numUser>1</cisco:numUser>
  <os:totalResults>10</os:totalResults>
  <os:itemsPerPage>2</os:itemsPerPage>
  <os:startIndex>0</os:startIndex>
  <os:Query cisco:filterBy="information" cisco:filterValue="imageGallery"
cisco:filterDisplayName="imageGallery" count="2" cisco:searchMetadata="true" role="subset"
searchTerms="pat" />
  <os:Query cisco:filterBy="information" cisco:filterValue="postAttachment"
cisco:filterDisplayName="postAttachment" count="1" cisco:searchMetadata="true"
role="subset" searchTerms="pat" />
  <os:Query cisco:filterBy="information" cisco:filterValue="post"
cisco:filterDisplayName="post" count="4" cisco:searchMetadata="true" role="subset"
searchTerms="pat" />
  <os:Query cisco:filterBy="information" cisco:filterValue="micropost"
cisco:filterDisplayName="micropost" count="2" cisco:searchMetadata="true" role="subset"
searchTerms="pat" />
  <os:Query cisco:filterBy="communities" cisco:filterValue="8510031"
cisco:filterDisplayName="test pat open community" count="3" cisco:searchMetadata="true"
role="subset" searchTerms="pat" />
  <os:Query cisco:filterBy="users" cisco:filterValue="10467" cisco:filterDisplayName="Pat
Kim" count="10" cisco:searchMetadata="true" role="subset" searchTerms="pat" />
  <os:Query cisco:filterBy="fileTypes" cisco:filterValue="gif"
cisco:filterDisplayName="gif" count="2" cisco:searchMetadata="true" role="subset"
searchTerms="pat" />
  <os:Query cisco:filterBy="fileTypes" cisco:filterValue="jpg"
cisco:filterDisplayName="jpg" count="1" cisco:searchMetadata="true" role="subset"
searchTerms="pat" />
  <entry>
    <id>10467</id>

```

```

<author>
  <cisco:id>10467</cisco:id>
  <email>pat@quad.cisco.com</email>
  <uri>/users</uri>
  <cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
</author>
<summary type="text">&lt;em>Pat&lt;/em> Kim
&amp;lt;pat@quad.cisco.com&amp;gt;</summary>
<updated>2012-06-28T09:48:46.000Z</updated>
<title type="text">&lt;em>pat&lt;/em> Kim</title>
<cisco:score>1.0</cisco:score>
<category term="users" />
<cisco:tags>rest</cisco:tags>
<cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
<link href="/users/10467" />
</entry>
<entry>
  <id>16110047</id>
  <author>
    <cisco:id>10467</cisco:id>
    <uri>/users</uri>
    <cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
  </author>
  <summary type="text">
</summary>
<updated>2012-06-13T12:01:40.000Z</updated>
<published>2012-06-13T12:01:40.000Z</published>
<title type="text">Orange.gif</title>
<cisco:score>0.015599317</cisco:score>
<category term="information" />
<cisco:resource>http://quad.cisco.com/schema/1.0/attachment</cisco:resource>
<link href="/attachments/16110047" />
</entry>
</feed>

```

Example 3 Request

```

GET http://webexsocialserver/api/quad/rest/search?q=test&searchMetadata=true&count=2
Accept:application/atom+xml

```

Example 3 Response

```

<feed xmlns="http://www.w3.org/2005/Atom" xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
xmlns:cisco="http://quad.cisco.com">
  <title type="text">http://quad.cisco.com OpenSearch</title>
  <updated>2012-06-28T16:55:45.340Z</updated>
  <author>
    <name>http://quad.cisco.com</name>
  </author>
  <cisco:totalInfo>1</cisco:totalInfo>
  <cisco:totalComm>0</cisco:totalComm>
  <cisco:totalUser>0</cisco:totalUser>
  <cisco:numInfo>1</cisco:numInfo>
  <cisco:numComm>0</cisco:numComm>
  <cisco:numUser>0</cisco:numUser>
  <os:totalResults>1</os:totalResults>
  <os:itemsPerPage>1</os:itemsPerPage>
  <os:startIndex>0</os:startIndex>
  <os:Query cisco:filterBy="information" cisco:filterValue="post"
cisco:filterDisplayName="post" count="1" cisco:searchMetadata="true" role="subset"
searchTerms="test" />
  <os:Query cisco:filterBy="users" cisco:filterValue="10467" cisco:filterDisplayName="Pat
Kim" count="1" cisco:searchMetadata="true" role="subset" searchTerms="test" />
</entry>

```

```
<id>8100003</id>
<author>
  <cisco:id>10467</cisco:id>
  <uri>/users</uri>
  <cisco:resource>http://quad.cisco.com/schema/1.0/user</cisco:resource>
</author>
<summary type="text">asdf</summary>
<updated>2012-06-13T12:01:40.000Z</updated>
<published>2012-06-13T11:59:37.000Z</published>
<title type="text">&lt;em>test</em> post</title>
<cisco:score>1.0</cisco:score>
<category term="information" />
<cisco:resource>http://quad.cisco.com/schema/1.0/post</cisco:resource>
<link href="/posts/8100003" />
</entry>
</feed>
```

