

Social Studio API Developer's Guide

Version 56.0, Spring '23



'23

CONTENTS

GETTING STARTED WITH THE SOCIAL STUDIO API	1
Chapter 1: Introducing Social Studio REST API	1
Enable API Applications for your Organization	2
Authentication	5
Authorization Code	6
Requesting an Access Token (Resource Owner Password)	7
Make Requests	8
Refresh the Access Token	9
Revoking an Access Token	10
GENERAL DEVELOPER INFO	11
Chapter 2: Rate Limiting	11
Response Headers	12
Chapter 3: Media Retrieval	13
METHODS	15
Chapter 4: Authors	15
Retrieve Authors	17
Create Author Tag	18
Delete Author Tag	20
Delete Author Tags	20
Create Comment for Author	21
Retrieve Comment for Author	22
Delete Author Comment	23
Chapter 5: Cases	25
Retrieve Cases	26
Update Cases	33
Chapter 6: Clients	35
Retrieve Clients	36
Chapter 7: Users	38
Retrieve Users	39
Retrieve User	40
Create User	41
Update User	43

Contents

Reset Password	44
Modify Password	45
Chapter 8: Workspaces	46
Retrieve Workspaces	48
Retrieve Workspace	50
Create Workspace	54
Update Workspace	55
Delete Workspace	57
Create Workspace Resources	59
Delete Workspace Resources	60
Chapter 9: Imported Posts	62
Retrieve Posts & Metrics	63
Chapter 10: Shared Content	72
Retrieve Shared Content	75
Share & Unshare Content	79
Create Shared Content	80
Archive & Unarchive Shared Content	82
Delete Shared Content	83
Chapter 11: Short URLs	85
Retrieve Short URLs	86
Retrieve Short URL	88
Create Short URL	89
Delete Short URL	91
Chapter 12: Publish Macros	93
Retrieve Publish Macros	94
Retrieve Publish Macro	96
Retrieve Publish Macro Attribute	97
Create Publish Macro	98
Delete Publish Macro	100
Chapter 13: Approval Rules	102
Retrieve Approval Rules	104
Retrieve Approval Rule	107
Create Approval Rule	112
Update Approval Rule	115
Delete Approval Rule	117
Retrieve Approval Rule Criteria	119
Update Approval Rule Criteria	123
Retrieve Approvers for a Rule	126
Update Approvers for a Rule	129

Chapter 14: Social Properties	133
Retrieve Properties	134
Retrieve Property	136
Chapter 15: Create Topic Profile and Retrieving Posts with Keyword Groups	138
Retrieve Topic Profiles	140
Retrieve Topic Profile	142
Create Topic Profile	143
Update Topic Profile	145
Delete Topic Profile	146
Update Keyword Groups in a Topic Profile	147
Delete Keyword Groups in a Topic Profile	148
Update Source Filters in a Topic Profile	148
Activate a Topic Profile	149
Chapter 16: Keyword Groups	151
Retrieve Keyword Groups	152
Create Keyword Groups	154
Update Keyword Groups	156
Chapter 17: Sources and Source Groups	158
Retrieve Sources	159
Retrieve Source	160
Create Source	161
Update Source	162
Delete Source	163
Retrieve Source Groups	164
Retrieve Source Group	165
Create Source Group	167
Update Source Group	168
Delete Source Group	169
Chapter 18: Languages	170
Retrieve Languages	171
Retrieve Language	172
Chapter 19: Regions	174
Retrieve Regions	175
Retrieve Region	176
Chapter 20: Media Types	178
Retrieve Media Types	179
Retrieve Media Type	180
Chapter 21: Posts	182

Retrieve Posts	183
Chapter 22: Engage Macros	190
Retrieve Macros	192
Retrieve Macro	194
Create Macro	196
Update Macro	197
Execute Macro	199
Delete Macro	200
Retrieve SalesforceOrgs	201
Retrieve SalesforceOrg	202
Chapter 23: Radian6 API Appendix	205
Supported Browsers	206
Supported Editions	206
Quick Start	206
Step One: Authenticate with the API	207
Step Two: Issue a Call to a Method	208
Step Three: Fetch Data	209
Using the Services	211
Post Service	211
User Service	214
Insight Service	216
Topic Service	218
Data Service	221
Blog Service	225
Authentication Service	228
Lookup Service	229
Resources	232

GETTING STARTED WITH THE SOCIAL STUDIO API

CHAPTER 1 Introducing Social Studio REST API

In this chapter ...

- [Enable API Applications for your Organization](#)
- [Authentication](#)

Available in: All editions.

The Social Studio API is a REST-ful web service for retrieving, analyzing, and modifying social media data contained in your Social Studio Account.

Clients may use the Social Studio API to extend the functionality of the Social Studio. For example: extract post data & insights directly from your Social Accounts and Topic Profiles, access post metadata such as post tags and source tags, and manage the setup and configuration within your Social Studio Organization.

The API may be used to integrate with any internal system, CRM, or Business Intelligence tool. Clients may use it to create custom internal reporting, to integrate with intranets, and to quickly disseminate information throughout the organization. Other uses include creating custom reports for clients, enhancing the value of client applications and services, or to drive web and mobile applications.

- RESTful API using HTTP verbs
- Authenticate using OAuth 2.0
- HTTPS only
- API requests and responses serialized as JSON
- Provides access to Social Studio functionality

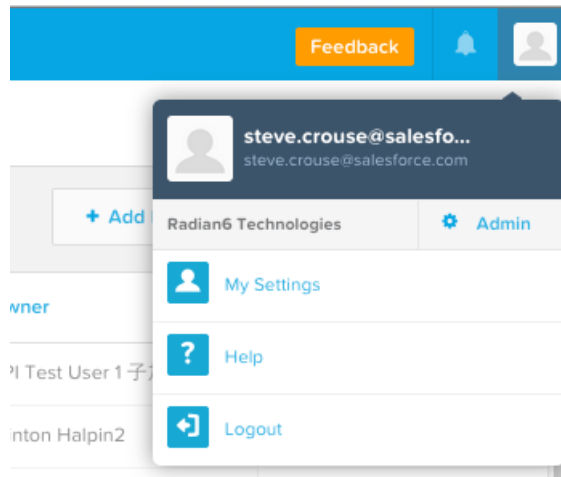
 **Note:** Social Studio API access can be used only for integration with other Salesforce systems or with other internal systems owned and operated by the customer. Salesforce sometimes monitors API usage. Uptime commitments or service-level agreements don't apply to API access. For more information, see the Social Studio Trust and Compliance documentation.

Enable API Applications for your Organization

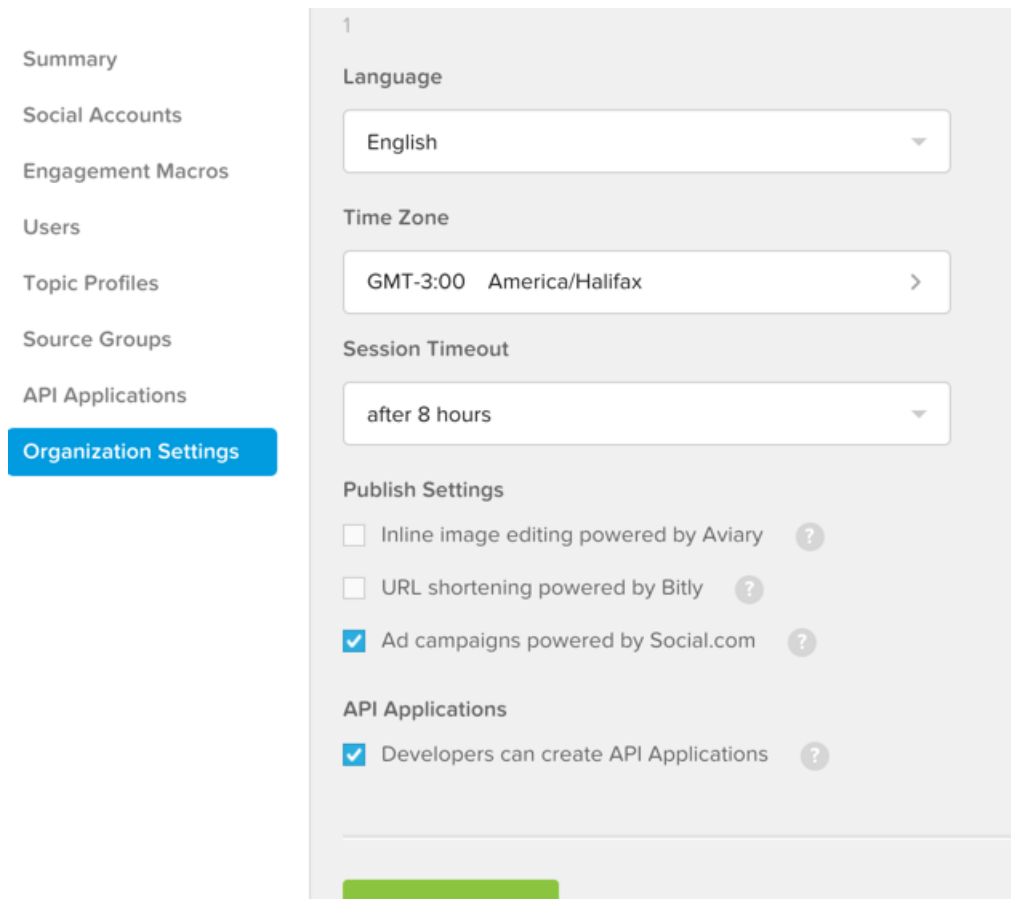
Before you utilize the Social Studio REST API, you must enable and register your API application:

This step ensures your organization can accept API calls.

1. Log in as a super user or organization administrator to <https://socialstudio.radian6.com/admin#organizationSettings>.
2. Navigate to the *Admin* section and click on *Organization Settings*.



3. Select the *Developers can create API Applications* checkbox under the *API Applications* heading.



1

Summary

Social Accounts

Engagement Macros

Users

Topic Profiles

Source Groups

API Applications

Organization Settings

Language

English

Time Zone

GMT-3:00 America/Halifax

Session Timeout

after 8 hours

Publish Settings

☐ Inline image editing powered by Aviary ?

☐ URL shortening powered by Bitly ?

☒ Ad campaigns powered by Social.com ?

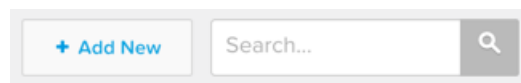
API Applications

☒ Developers can create API Applications ?

Register an API Application in Social Studio

This step creates a new application for use with Social Studio and provides an application key and secret for use in retrieving an OAuth access token.

1. Navigate to the *Admin* section in Social Studio.
2. Click on the *API Applications* section.
3. Click *Add New*.



+ Add New

Search...

4. Review the Marketing Cloud Social API Usage Agreement and click *I Accept the Terms and Conditions*.

Marketing Cloud Social API Usage Agreement

This is a legally binding agreement. Please read it carefully. By clicking "I Agree," or installing or using the API - Radian6 and Social Studio (the "API"), you:

- agree to the following terms on behalf of the Radian6 or Social Studio customer with which you are employed, affiliated or associated, (the "Customer"),
- represent that you have the authority to bind the Customer to these terms, and
- represent that you are an authorized User under the master subscription agreement between salesforce.com and the Customer (the "Master Subscription Agreement").

If you do not have such authority, are not an authorized User, or do not agree to these terms, you may not install or use the API .

This Order Form Supplement was last updated on August 16, 2015. It is effective between the Customer and salesforce.com as of the date you first download, install or use the API, whichever is earliest.

1. THIS ORDER FORM SUPPLEMENT

This Order Form Supplement is a part of the order form, ("Order Form") pursuant to which the Customer purchased salesforce.com Services branded as "Radian6" or "Social

CancelI Accept the Terms and Conditions

5. Enter the details for your new API application: *Name Application Description Callback URL*

New API Application

Name

Application Description (optional)

Callback URL (optional)

6. Click Save.
7. To view the details of your application, find it in the applications list and click on it. A dialog will appear on the left side of your screen that contains all of your application details.

Details



Tim Application
<script>alert('hi');stage3</script>



Edit



Delete

Name
Tim Application

Application Description
test

Callback URL

Application Key


Application Secret

Show Secret

Reset Application Secret

Authentication

You can receive the client key and secret issued for the application via self-serve by a Social Studio super user or an administration by logging into socialstudio.radian6.com/admin. When creating the credentials for the application, you must specify the redirect URL for the application. The authentication server will redirect the user to this URL with an authorization code.

Supported Flows

Social Studio currently supports the following OAuth flows:

- Authorization (web server flow)
- Implicit grant (client-side applications)
- Resource owner password-based grant (trusted applications)

Authorization Code

This authentication method redirects the user to an authorization page, which permits the user to grant authorization to the application to act. The page displays a login prompt with the message *Please log in to grant access to the application*. Once the user logs in and grants access, the page will redirect the user to the application redirect URL as specified when generating their client credentials.

Authorization URL: <https://api.socialstudio.radian6.com/login/oauth/authorize>

Request

Query String Parameter	Value
client_id	The application client key
response_type	Either one of the following two elements: <i>code</i> - Server-side applications use this web service flow. After authentication, the browser redirects back to the application redirect URL with an authorization code on the query string used to request a refreshable access token. <i>token</i> - Client-side applications use this implicit (client-credentials) flow. This process does not store an access token. Instead, the call returns the token as part of the URL fragment.
state	A unique value included in the redirect call that maintains the state between the request and the response
redirect_uri	

Response

The call returns the response of this endpoint to the client application by redirecting back to their returned URL with certain query string parameters set.

Web Server Flow

After the user successfully logs in, the call redirects them to the following URL:

```
{redirectUri}?code={authorizationCode}&scope=OOB&state={state}
```

redirectUri - the call specifies this value when creating the client credentials.
authorizationCode - the application can handle the authorizationCode as outlined in the section **Requesting an Access Token**.
state - use the *state* value you specified in your requests.
scope - at present, we do not support specific scopes, so use the default value of OOB

Requesting an Access Token (Resource Owner Password)

This flow includes implied authorization because the resource owner conducts the call. Use this flow only for highly trusted clients, as the process exposes the user password to the client. You only need to send the password once to receive a refresh token. Store the refresh token for further requests.

Request

```
POST /oauth/token HTTP/1.1
Host: api.socialstudio.radian6.com
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=password&client_id={clientKey}&client_secret={clientSecret}&username={username}&password={password}
```

```
*authorizationCode - code returned by following the instructions in the *Authorization
Code* section.
*client_id - client key included in client credentials for the application
*client_secret* - client secret included in the client credentials for the application
*username* - username in Social Studio
*password* - password in Social Studio
```

cURL Example

```
curl -X POST -H "Cache-Control: no-cache" -H "Content-Type:
application/x-www-form-urlencoded" -d
'grant_type=password&client_id=your_marketing_cloud_api_key&client_secret=your_marketing_cloud_api_secret&username=your_username&password=your_password'
'https://api.socialstudio.radian6.com/oauth/token'
```

Response

```
{
  "access_token": "bc0fb4f7-2f51-4aa6-9192-f559e481e02f",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "d0e05be9-fb84-4dfb-93a5-8d781024cdc8",
  "scope": "oob"
}
```

```
*access_token* - value used to make requests (as shown in the *Make Requests* section)
*expiresIn* - number of seconds before the token expires
*refresh_token* - value to save and use to refresh the access token (as shown in the
*Refresh the Access Token* section)
```

Make Requests

You can use a new API host that supports OAuth:

```
api.socialstudio.radian6.com
```

Requests to this URL require only an access token and do not require an *auth_token* or *auth_appkey* value.

Request

```
GET {host}v3/user/me HTTP/1.1
Host: api.socialstudio.radian6.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/json;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
access_token: 5545b2f1-9a8e-4bdd-a2ee-9d0c70f8fb08

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Request-ID: IsZ0g0ryksn0Cvk6
Content-Encoding: gzip
Content-Type: application/json
Content-Length: 156
Date: Tue, 18 Jun 2013 19:49:02 GMT

{
  "data": [
    {
      "id": "7914",
      "username": "john.doe@domain.com",
      "displayName": "John Doe",
      "email": "john.doe@domain.com",
      "timeZone": "America/Halifax",
      "enabled": true,
      "orgRoleId": 1,
      "languageId": 1,
      "createdDate": "2014-10-27T14:41:48Z",
      "internalUserAvatarUrl": null,
      "userAvatarUrl": null,
      "clientId": 1
    }
  ],
  "meta": {
    "totalCount": 1
  }
}
```

Alternatively, you can supply the access token using the header (188+):

```
Authorization: Bearer {access_token}
```

cURL Example

```
curl -X GET -H "access_token: your_access_token" -H "Cache-Control: no-cache"  
'https://api.socialstudio.radian6.com/v3/users/me'
```

Refresh the Access Token

The Social Studio API will respond with the following message after token expiration, revocation, or any other issue leading to an invalid token:

```
HTTP/1.1 401 Unauthorized  
Date: Wed, 04 Nov 2015 18:07:59 GMT  
Server: Apache-Coyote/1.1  
Content-Type: application/json;charset=UTF-8  
Content-Length: 279  
  
{  
  "error": {  
    "message": "Validation error",  
    "statusCode": 401,  
    "errorCode": "MC-Unknown",  
    "requestId": "9a135ab7-9e56-49d7-8c1d-7805f2df6ee1"  
  },  
  "meta": null  
}
```

An application can wait for this error message and then attempt to refresh the token, or it can use the `expires_in` time that comes with the access token to determine when the token needs to be refreshed. The token can be refreshed at any time.

The end point for refreshing an access token is the same as that for requesting an access token, but uses slightly different parameters.

```
api.socialstudio.radian6.com/oauth/token
```

Request

```
POST /oauth/token HTTP/1.1  
Host: api.socialstudio.radian6.com  
Content-Length: 169  
Content-Type: application/x-www-form-urlencoded  
Accept: */*  
Accept-Encoding: gzip, deflate, sdch  
  
grant_type=refresh_token&refresh_token={refreshToken}&client_id={clientKey}&client_secret={clientSecret}
```

In this case, the `grant_type` is `refresh_token` and instead of supplying a code the parameter is `refresh_token`.

Response

If the refresh token has been revoked by the user or has become invalid for some other reason, then the refresh call will fail with the following error:

```
HTTP/1.1 400 Bad Request
Date: Wed, 04 Nov 2015 18:12:16 GMT
Server: Apache-Coyote/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 78
Connection: close

{
  "error": "invalid_request",
  "error_description": "Invalid refresh token"
}
```

Revoking an Access Token

The user can revoke the access token and the refresh token using the *access_token* value.

Request

```
GET /oauth/revoke?access_token={access_token} HTTP/1.1
Host: api.socialstudio.radian6.com
Accept: */*
Accept-Encoding: gzip, deflate, sdch
```

Response

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Encoding: gzip
Content-Type: text/xml;charset=UTF-8
Content-Length: 31
Date: Wed, 07 Aug 2013 18:29:04 GMT

<revoked />
```


GENERAL DEVELOPER INFO

CHAPTER 2 Rate Limiting

In this chapter ...

- [Response Headers](#)

The Social Studio API will subject users who make too many calls too quickly to rate limiting. This measure helps to ensure a single misbehaving application cannot produce excessive load and cause a degradation in performance for other applications.

Rate limiting breaks calls to the API into two types:

*Relatively expensive calls that need to scan large indexes can make 700 requests per hour per user. GET /v3/posts. *Less expensive API calls can make 900 requests per hour per user.*

Response Headers

The Social Studio API sends additional headers containing rate limit information with each response. These headers help track application usage and quotas.

```
*X-RateLimit-Limit* - limit for the current endpoint type
*X-RateLimit-Remaining* - number of requests remaining for the current endpoint type
in the current timeframe
*X-RateLimit-Reset* - number of seconds until the limit period expires and the API
resets the limit to the number found in the *X-RateLimit-Limit* header
```

CHAPTER 3 Media Retrieval

To retrieve images and video from the Social Studio API a JSON Web Token must be created and passed along with the request.

JSON Web Tokens consist of three parts separated by a dot (.), which are the Header, Payload, and Signature.

Header

The header is a URL-safe Base64 encoding describing the hashing algorithm. In the case of Social Studio it is:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload

The payload contains the claims, where are also URL-safe Base64 encoded.

```
{
  "request": {
    "userId": "2999",
    "clientId": "1"
  },
  "iat": 1458136904,
  "client_id": "9283009918837749",
  "aud": "http://www.radian6.com",
  "sub": "layer7",
  "exp": 1489672904
}
```

```
{
  *userId* - your Social Studio user id
  *clientId* - your Social Studio organization id, this can
be found under Organization Settings within Social Studio.
```

```
*iat* - date in epoch seconds the JWT was generated
*client_id* - client of the application that was created
within Social Studio
*exp* - expiration time, in epoch seconds.
```

Signature

To create the signature, combine the encoded header, encoded payload, your access token, the algorithm in the header and sign that.

```
HMACSHA256(
    base64URLEncode(header) + "." +
    base64URLEncode(payload),
    access_token)
```

Put it all together

The output is three Base64 encoded strings separated by dots: header.payload.signature. When you make a request that includes images or video specify the JWT as a token query parameter as part of the URL.

```
GET /v1/clips/{id}?token="header.payload.signature"
```

METHODS

CHAPTER 4 Authors

In this chapter ...

- [Retrieve Authors](#)
- [Create Author Tag](#)
- [Delete Author Tag](#)
- [Delete Author Tags](#)
- [Create Comment for Author](#)
- [Retrieve Comment for Author](#)
- [Delete Author Comment](#)

The Authors endpoint allows you to get detailed information on an author. You can also apply additional metadata by adding and managing tags and notes associated with those authors. Use author tags to identify VIP customers, promoters, or detractors of a brand. Use notes to record additional important information on an author record.

Retrieve an Author

You can retrieve information on an author using the following call:

```
GET /v3/authors/{id}
```

cURL Example

```
curl -X GET -H "access_token: 0682240d-f531-4dc1-a9a6-c80eba9cb533"
'https://api.socialstudio.radian6.com/v3/authors/36221001'
```

Retrieve Author Tags

Add author tags using the following call:

```
POST /v3/authors/{id}/tags
```

cURL Example

```
curl -X POST -H "access_token: 0682240d-f531-4dc1-a9a6-c80eba9cb533"
-H "Content-Type: application/json" -d '{
  "value": "VIP"
}' 'https://api.socialstudio.radian6.com/v3/authors/36221001/tags'
```

Delete Author Tags

Delete author tags using the following call:

```
DELETE /v3/authors/{id}/tags
```

cURL Example

```
curl -X DELETE -H "access_token: 0682240d-f531-4dc1-a9a6-c80eba9cb533"
-H "Content-Type: application/json" -d '{
"value":"VIP"
}'
'https://api.socialstudio.radian6.com/v3/authors/36221001/tags/6265'
```

Retrieve Authors

This method retrieves the specified author.

```
/v3/authors/{id}
```

HTTP method

```
GET
```

Request Parameters

id - string value indicating the author of a post (required)

Request Parameter Example

```
GET /v3/authors/{id}
```

Sample Request

```
GET /v3/authors/{id}
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "2184",
      "title": "R6APITestUser1",
      "externalId": "429245766",
      "externalLegacyId": null,
      "authorFullName": "API 1 APIQE □□□□□□",
      "avatar": "http://pbs.twimg.com/profile_images/1815149549/b-410745-
animated_dog_normal.jpg",
      "authorTags": [
        {
          "id": "13434",
          "value": "Sample tag"
        }
      ],
      "authorComments": [
        {
          "id": "454545",
          "value": "Sample note"
        }
      ]
    }
  ]
}
```

```

    },
    "bio": "Sample bio",
    "verified": false,
    "authorData": {
      "blogId": "62550515"
    }
  },
  "meta": {
    "totalCount": 1
  }
}

```

id - string value indicating author
 title - string value indicating author title
 externalid - string value indicating ID for author on external platform
 externalLegacyId - string value indicating any other external platform identifier for author
 authorFullName - string value indicating any available full name for author
 avatar - string value indicating URL for avatar image file
 authorTags - array containing ID and value pairs for tags associated with author
 authorComments - array containing ID and value pairs for comments associated with author
 bio - string value containing bio information for author from external platform
 verified - boolean value indicating verified status for Twitter
 authorData - array containing additional information on author, such as blogID

A call referencing a nonexistent author returns a 404 error.

A call that encounters an internal error while retrieving the author returns a 500 error.

Create Author Tag

This method adds a tag value to an author object within your Social Studio organization. You can add a single tag in a call or pass an array of multiple tag values

```
/v3/authors/{authorId}/tags
```

HTTP Method

```
POST
```

Request Parameters

authorId - Integer value indicating the ID of the specific author (required)

Request Parameters Example

```
POST /v3/authors/{authorId}/tags
```


JSON Parameters

`*value*` - string value including the tag value to apply to the author

Sample Request

The call below passes a single tag to the author.

```
POST /v3/authors/1/tags
{
  "value": "vino"
}
```

The call below passes an array of tags to the author.

```
POST /v3/authors/1/tags
[
  {
    "value": "Some Tag Value 1"
  },
  {
    "value": "Some Tag Value 2"
  },
  {
    "value": "Some Tag Value 3"
  }
]
```

Sample Response

A successful call returns a 201 code with a Location header for the location of the new tag and an ID value. If you pass an array of tags, that response will include a Location header and an array of IDs for the tags.

```
[
  {
    "id": "154"
  },
  {
    "id": "155"
  },
  {
    "id": "156"
  }
]
```

A call including an invalid JSON object will return a 400 code. Ensure you correctly formed the JSON object.

A call referencing an invalid author ID will return a 404 code. Ensure you referenced the correct author value.

A duplicate tag returns a 409 code. Ensure you include a unique tag value in your call.

Delete Author Tag

This method deletes a tag value for an author object within your Social Studio organization.

```
/v3/authors/{authorId}/tags/{tagId}
```

HTTP Method

```
DELETE
```

Request Parameters

authorId - Integer value indicating the ID of the specific author (required)
tagId - Integer value indicating the ID of the specific tag (required)

Request Parameters Example

```
DELETE /v3/authors/{authorId}/tags/{tagId}
```

Sample Request

```
DELETE /v3/authors/12345/tags/123
```

Sample Response

A successful call returns a 204 code.

A call referencing an invalid author ID or tag ID will return a 404 code. Ensure you referenced the correct values.

A internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Delete Author Tags

This method deletes all tag values associated with an author object within your Social Studio organization.

```
/v3/authors/{authorId}/tags
```

HTTP Method

```
DELETE
```

Request Parameters

`*authorId*` - Integer value indicating the ID of the specific author (required)

Request Parameters Example

```
DELETE /v3/authors/{authorId}/tags
```

Sample Request

```
DELETE /v3/authors/12345/tags
```

Sample Response

A successful call returns a 204 code.

A call that could not remove all tags from an author returns a 400 code.

A call referencing an invalid author ID will return a 404 code. Ensure you referenced the correct author value.

An internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Create Comment for Author

This method creates a comment for the specified author.

```
/v3/authors/{id}/comments
```

HTTP Method

```
POST
```

Request Parameters

`*id*` - string value indicating the author of a post (required)

Request Parameters Example

```
POST /v3/authors/{id}/comments
```

Sample Request

```
POST /v3/authors/{id}/comments
{
```

```
{
  "value": "i heart talk radio"
}
```

Sample Response

A successful call returns a 201 code with the applicable Location header.

A call including an invalid JSON object returns a 400 code.

A call including an invalid author returns a 404 code.

A call encountering an internal server error while adding the comment returns a 500 code.

Retrieve Comment for Author

This method returns all comments for a specified author.

```
/v3/authors/{id}/comments
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - string value indicating the author of a post (required)
*limit* - integer value indicating number of results to return (defaults to 25). This
method does not return any deleted comments.
*offset* - integer value indicating result to start with (defaults to 0)
```

Request Parameter Example

```
GET /v3/authors/{id}/comments
```

Sample Request

```
GET /v3/authors/{id}/comments
```

Sample Request

A successful call returns the following response:

```
{
  "data" : [
    {
      "id": "2751",
      "authorId": 53,
```

```

    "value": "i heart talk radio",
    "deleted": null,
    "created": {
      "userId": 26341,
      "clientId": 11084,
      "name": "APIQECLIENT1USER1",
      "date": "2015-07-27T13:53:48Z"
    }
  },
  "meta" : {
    "totalCount" : 1
  }
}

```

```

*data* - array of information on comment
*id* - string value identifying comment
*authorId* - string value identifying author of comment
*value* - string value containing text of comment
*deleted* - string value indicating a deleted comment
*created* - array of information about comment creation
*userId* - string value of user ID for comment within system
*clientId* - string value of client ID for comment within system
*name* - string value of comment within system
*date* - timestamp value of creation within system

```

Sample Response

A call referencing a nonexistent author returns a 404 error.

A call that encounters an internal error while retrieving the author returns a 500 error.

Delete Author Comment

This method deletes a comment from the specified author.

```
/v3/authors/{id}/comments/{commentId}
```

HTTP Method

```
DELETE
```

Request Parameters

```

*id* - string value indicating the author of a post (required)
*commentId* - string value indicating the comment to delete (required)

```

Request Parameters Example

```
DELETE /v3/authors/{id}/comments/{commentId}
```

Sample Request

```
DELETE /v3/authors/{id}/comments/{commentId}
```

Sample Response

A successful call returns a 204 No Content code.

A call including an invalid author, a previously deleted note, or a nonexistent note returns a 404 code.

A call encountering an internal server error while deleting the comment returns a 500 code.

CHAPTER 5 Cases

In this chapter ...

- [Retrieve Cases](#)
- [Update Cases](#)

Cases are opened as a result of Approval Rules within a Workspace. Within Social Studio you may see cases represented in the Workspace's Tasks page. The Workspace Tasks page displays active cases within the workspace where the logged in user is the approver and where the approvalstatus is pending. You may want to use this end-point along with GET /workspaces to build a cross-workspace Tasks page.

Retrieve Cases

This method retrieves a collection of cases.

```
/v1/cases
```

HTTP method

```
GET
```

Request Parameters

```
*workspace* - string value indicating ID of workspace
*key* - string value indicating name of key used to filter case (can include multiple
pairs)
  *status* - string value indicating status of cases
    *active* - string value indicating cases in approval process
    *rejected* - string value indicating cases rejected during approvals process
    *approved* - string value indicating cases approved during approvals process
    *canceled* - string value indicating cases canceled during approvals process due
to another case opening
  *producer* - numeric value indicating producer who initiated cases
  *approver* - numeric value indicating cases requiring approval from a specified user
  *approvalstatus* - string value indicating approval status of cases (requires approver
key as well), can include comma-separated values
    *pending* - string value indicating pending cases
    *approved* - string value indicating approved cases
  *itemtype* - string value indicating cases initiated on a specific item type, can
include comma-separated values
    *post* - string value indicating cases initiated by post
  *item* - string value indicating cases initiated on a specific item (requires itemtype
key as well)
  *trigger* - string value indicating action that initiated case, can include
comma-separated values
    *publish* - string value indicating cases triggered by publish (default)
    *promotion* - string value indicating cases triggered by promotion
  *hydrate* - string value indicating inclusion of additional information on specified
value instead of default value
    *approvers* - string value indicating hydration on approver key and value
    *producer* - string value indicating hydration on producer key and value
    *modifiedby* - string value indicating hydration on modifiedby key and value
  *include* - string value indicating additional fields to include in response, can
include comma-separated values
    *approvers* - array value indicating approvers and statuses associated with case
  *progress* - array value indicating current progress step and total number of steps
involved
  *sorting* - string value indicating sorting for response results
  *updated* - string value indicating most recent update (status change or approver
action)
  *created* - string value indicating most recently created cases (case first moves
```



```

to active status)
  *direction* - string value indicating direction of response results
    *asc* - string value indicating ascending order of response results
    *desc* - string value indicating descending order of response results
  *perpage* - numeric value indicating number of response results to return per page
  *page* - numeric value indicating page number to start displaying results
  *value* - string value indicating value of key used to filter case (can include multiple
pairs)

```

Request Parameter Example

```
GET /v1/cases?workspace=value&key1=value1&...keyN=valueN
```

Sample Request - Case by ID

```
GET /v1/cases/95b75fac-0453-11e3-8d6f-168170973bd
```

Sample Response - Case by ID

A successful call returns the following response:

```

{
  "status": true,
  "response": {
    "id": "95b75fac-0453-11e3-8d6f-168170973bd",
    "created": "2013-10-30 16:41:12",
    "updated": "2013-10-30 18:41:12",
    "status": "active",
    "itemtype": "post",
    "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
    "producer": 456,
    "modifiedby": 789,
    "trigger": "publish"
  },
  "meta": []
}

```

Sample Request - Case by ID with includes

```
GET /v1/cases/95b75fac-0453-11e3-8d6f-168170973bd?include=progress,approvers
```

Sample Response - Case by ID with includes

A successful call returns the following response:

```

{
  "status": true,
  "response": {
    "id": "95b75fac-0453-11e3-8d6f-168170973bd",

```

```

    "created": "2013-10-30 16:41:12",
    "updated": "2013-10-30 18:41:12",
    "status": "active",
    "itemtype": "post",
    "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
    "producer": 456,
    "modifiedby": 789,
    "trigger": "publish",
    "approvers": [
      {
        "id": 123,
        "approvalstatus": "pending",
        "completed": "1970-01-01 00:00:00"
      }
    ],
    "progress": {
      "step" : "2",
      "total_steps" : "10"
    }
  },
  "meta": []
}

```

Sample Request - Case by ID with hydration

```

GET
/v1/cases/95b75fac-0453-11e3-8d6f-168170973bd?include=progress,approvers&hydrate=approvers,producer,modifiedby

```

Sample Response - Case by ID with hydration

A successful call returns the following response:

```

{
  "status": true,
  "response": {
    "id": "95b75fac-0453-11e3-8d6f-168170973bd",
    "created": "2013-10-30 16:41:12",
    "updated": "2013-10-30 18:41:12",
    "status": "active",
    "itemtype": "post",
    "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
    "producer": {
      "id": "456",
      "email": "myuser@domain.com",
      "title": "My User",
      "avatar_url":
"http://socialstudio.s3.amazonaws.com/avatars/bde3c2170bf9bbeaf3038f78b042cd10",
      "timezone": "America/New_York",
      "enabled": false,
      "updated": "2014-01-21T21:35:45+0000",
      "organization_id": "10567",
      "roles": [

```

```

        "ROLE_ORGANIZATION_SUPER_USER.10567"
    ]
},
"modifiedby": {
    "id": "789",
    "email": "anotheruser@domain.com",
    "title": "Another User",
    "avatar_url":
"http://socialstudio.s3.amazonaws.com/avatars/aae3c2170bf9bbeaf3038f78b042cd10",
    "timezone": "America/New_York",
    "enabled": false,
    "updated": "2014-01-21T21:35:45+0000",
    "organization_id": "10567",
    "roles": [
        "ROLE_ORGANIZATION_SUPER_USER.10567"
    ]
},
"trigger": "publish",
"approvers": [
    {
        "id": "123",
        "approvalstatus": "pending",
        "completed": "1970-01-01 00:00:00",
        "email": "approveruser@domain.com",
        "title": "Approver User",
        "avatar_url":
"http://socialstudio.s3.amazonaws.com/avatars/cce3c2170bf9bbeaf3038f78b042cd10",
        "timezone": "America/New_York",
        "enabled": false,
        "updated": "2014-01-21T21:35:45+0000",
        "organization_id": 10567,
        "roles": [
            "ROLE_ORGANIZATION_SUPER_USER.10567"
        ]
    }
],
"progress": {
    "step": "0",
    "total_steps": "1"
}
},
"meta": []
}

```

Sample Request - Cases produced by workspace user

```
GET /v1/cases?workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&producer=456
```

Sample Response - Cases produced by workspace user

A successful call returns the following response:

```
{
  "status": true,
  "response": [
    {
      "id": "95b75fac-0453-11e3-8d6f-168170973bd",
      "created": "2013-10-30 16:41:12",
      "updated": "2013-10-30 18:41:12",
      "status": "active",
      "itemtype": "post",
      "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
      "producer": 456,
      "modifiedby": 789,
      "trigger": "publish"
    }
  ],
  "meta": {
    "total": 1,
    "page": 1,
    "perpage": 10,
    "current":
"https://api.com/v1/cases?producer=456&workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&page=1",

    "prev": "",
    "next": ""
  }
}
```

Sample Request - Case awaiting approval from workspace user

```
GET /v1/cases?workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&approver=123
```

Sample Response - Case awaiting approval from workspace user

A successful call returns the following response:

```
{
  "status": true,
  "response": [
    {
      "id": "95b75fac-0453-11e3-8d6f-168170973bd",
      "created": "2013-10-30 16:41:12",
      "updated": "2013-10-30 18:41:12",
      "status": "active",
      "itemtype": "post",
      "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
      "producer": 456,
      "modifiedby": 789,
      "trigger": "publish"
    }
  ],
}
```

```

    "meta": {
      "total": 1,
      "page": 1,
      "perpage": 10,
      "current":
"https://api.com/v1/cases?approver=123&workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&page=1",

      "prev": "",
      "next": ""
    }
  }
}

```

Sample Request - Case with information on a post

```

GET
/v1/cases?workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&itemtype=post&itemid=00004b3f-f0f0-47b3-a100-d3a2367fb440

```

Sample Response - Case with information on a post

A successful call returns the following response:

```

{
  "status": true,
  "response": [
    {
      "id": "95b75fac-0453-11e3-8d6f-168170973bd",
      "created": "2013-10-30 16:41:12",
      "updated": "2013-10-30 18:41:12",
      "status": "active",
      "itemtype": "post",
      "item": "00004b3f-f0f0-47b3-a100-d3a2367fb440",
      "producer": 456,
      "modifiedby": 789,
      "trigger": "publish"
    }
  ],
  "meta": {
    "total": 1,
    "page": 1,
    "perpage": 10,
    "current":
"https://api.com/v1/cases?workspace=00004b3f-f0f0-47b3-a100-d3a2367fb440&itemtype=post&itemid=00004b3f-f0f0-47b3-a100-d3a2367fb440&page=1",

    "prev": "",
    "next": ""
  }
}

```

Error - Invalid Filter

An invalid filter returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "{1st validation error}, ..., {nth validation error}"
      }
    ]
  },
  "meta": []
}
```

- workspace value not provided - "Workspace value cannot be blank, Workspace value cannot be null"
- workspace value empty - "Workspace value cannot be blank"
- workspace value does not include precisely 36 characters - "Invalid value for workspace"
- page value less than or equal to 0 - "This value should be greater than 0."
- page value exceeds last page containing items "The page number is exceeding the maximum allowed number."
- perpage value less than or equal to 0 - "This value should be greater than 0."
- approvalstatus value does not include a specified approver - "Invalid approver supplied"
- approvalstatus value includes unsupported values or no value - "One or more of the given values is invalid."
- itemtype value includes multiple comma-separated values and item includes a value - "Invalid itemtype passed. Should be exactly one."
- trigger value included unsupported values or an empty value - "One or more of the given values is invalid."

Error - Insufficient Permissions

A request with insufficient permissions returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "..."
      }
    ]
  },
  "meta": []
}
```

****workspace**** value includes a workspace user cannot view - "Permission denied on the passed workspace"

Update Cases

This method allows approvers to update the status of a case.

```
/v1/cases
```

HTTP method

```
PATCH
```

Request Parameters

```
*id* - string value indicating ID of case (required)
```

Request Parameter Example

```
PATCH /v1/cases/{id}
```

JSON Parameters

```
*approvalaction* - string value indicating approval action to take on case (required)
  *approve* - string value indicating approval of case
  *reject* - string value indicating rejection of case
*message* - optional message to include with approval action
```

Sample Request

```
PATCH /v1/cases/95b75fac-0453-11e3-8d6f-168170973bd
{"approvalaction": "approve", "message": "good"}
```

Sample Response

```
{
  "status": true,
  "response": true,
  "meta": []
}
```

Error - Invalid Parameter Supplied

An invalid parameter returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "...
      }
    ]
  },
  "meta": []
}
```

- approvalaction value not present or includes unsupported value - "An invalid parameter/value was supplied"
- id value does not include precisely 36 characters - "An invalid parameter/value was supplied"
- Submitting an approval on a case that does not include status with an active value - "Case has already been closed"
- Submitting an approval without a pending approver status - "Submission already received"

Error - Not able to submit an approval

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to submit approval"
      }
    ]
  },
  "meta": []
}
```


CHAPTER 6 Clients

In this chapter ...

- [Retrieve Clients](#)

Use the Clients endpoint to retrieve basic information about the Social Studio organization (such as org name, ID, timezone, and language).

Retrieve Client Information

Retrieve client information using the following call:

```
GET /v3/clients
```

cURL Example

```
curl -X GET -H "access_token: 82200099-7799-4023-b76d-11bb7c941bf1"  
'https://api.socialstudio.radian6.com/v3/clients'
```

Retrieve Clients

This method retrieves the Social Studio organization for your company. At this time, this call returns only the client for the authenticated user.

```
/v3/clients
```

HTTP method

```
GET
```

Request Parameters

```
None
```

Request Parameter Example

```
GET /v3/clients
```

Sample Request

```
GET /v3/clients
```

Sample Response

```
{
  "data": [
    {
      "id": "1",
      "title": "Radian6 Technologies",
      "email": null,
      "timeZone": "America/Halifax",
      "languageId": 1,
      "sessionTimeoutMillis": 28800000,
      "aviaryEnabled": false,
      "apiSelfServeEnabled": true
    }
  ],
  "meta": {
    "totalCount": 1
  }
}
```

```
*id* - integer value identifying organization
*title* - string value for name of organization
*email* - string value for email address associated with organization
```

```
*timeZone* - string value for time zone associated with organization
*languageId* - integer value indicating language associated with organization
*sessionTimeoutMillis* - integer value indication number of milliseconds associated with
timeout value for organization
*aviaryEnabled* - boolean value indicating whether organization can use aviary
*apiSelfServeEnabled* - boolean value indicating organization access to API self-service
features
*bitlyEnabled* - boolean value indicating whether Bitly is enabled
```

CHAPTER 7 Users

In this chapter ...

- [Retrieve Users](#)
- [Retrieve User](#)
- [Create User](#)
- [Update User](#)
- [Reset Password](#)
- [Modify Password](#)

Use the Users endpoint to manage basic aspects of Social Studio user accounts.

Retrieve a User

```
GET /v3/users/{id}
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e"
'https://api.socialstudio.radian6.com/v3/users/me'
```

Create a User

```
POST /v3/users
```

cURL Example

```
curl -X POST -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e"
-H "Content-Type: application/json" -d ' {
  "title": "API User",
  "username": "APIUser",
  "displayName": "API User",
  "email": "email@salesforce.com",
  "timeZone": "Europe/London",
  "enabled": true,
  "avatarURL": null,
  "userRoleId": 6,
  "orgRoleId": 1,
  "languageId": 1,
  "clientId": 1
}' 'https://api.socialstudio.radian6.com/v3/users'
```

Retrieve Users

This method retrieves information for all users within an account

```
/v3/users
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*q* - string value including characters to search for in title (minimum of 3 characters)
*enabled* - boolean value indicating whether to retrieve only active users
    *true* - string value indicating return only active users
    *false* - string value indicating return all users
*limit* - integer value indicating number of results to return (defaults to 100,000)
*offset* - integer value indicating result to start with (defaults to 0)
```

Sample Request

```
GET /v3/users
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "3339",
      "title": "Admin",
      "username": "admin",
      "displayName": "Admin",
      "email": "testuser@example.com",
      "timeZone": "America/Goose_Bay",
      "enabled": true,
      "orgRoleId": 1,
      "languageId": 1,
      "createdDate": "2014-05-09T03:00:00Z",
      "internalUserAvatarUrl": null,
      "userAvatarUrl": null,
      "clientId": 1
    },
    ...
  ]
}
```

```
  ],
  "meta": {
    "totalCount" : 52
  }
}
```

id - integer value indicating unique ID for user
title - string value including name fo users
username - string value including network username for user
email - string value including email address for user
timeZone - string value indicating location for user
enabled - boolean value indicating user status
 true - string value indicating enabled
 false - string value indicating disabled user
orgRoleId - integer value indicating role for user within organization
languageId - integer value indicating language used by user
createdDate - timestamp value indicating when system created user
internalUserAvatarUrl - string value indicating location of avatar image file for internal use
userAvatarUrl - string value indicating location of avatar image file
clientId - integer value indicating client ID for specific user

Retrieve User

This method retrieves information for a specified user within an account

```
/v3/users/{id}
```

HTTP Method

```
GET
```

Request Parameters

id - integer value indicating unique ID for user (you may also use the ***me*** shorthand in place of this value)

Sample Request

```
GET /v3/users/{id}
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
```

```

    "id": "3339",
    "title": "Admin",
    "username": "admin",
    "displayName": "Admin",
    "email": "testuser@example.com",
    "timeZone": "America/Goose_Bay",
    "enabled": true,
    "orgRoleId": 1,
    "languageId": 1,
    "createdDate": "2014-05-09T03:00:00Z",
    "internalUserAvatarUrl": null,
    "userAvatarUrl": null,
    "clientId": 1
  },
  ...
],
"meta": {
  "totalCount" : 1
}
}

```

id - integer value indicating unique ID for user
 title - string value including name fo users
 username - string value including network username for user
 email - string value including email address for user
 timeZone - string value indicating location for user
 enabled - boolean value indicating user status
 true - string value indicating enabled
 false - string value indicating disabled user
 orgRoleId - integer value indicating role for user within organization
 languageId - integer value indicating language used by user
 createdDate - timestamp value indicating when system created user
 internalUserAvatarUrl - string value indicating location of avatar image file for internal use
 userAvatarUrl - string value indicating location of avatar image file
 clientId - integer value indicating client ID for specific user

Create User

This method creates a new user within a Social Studio tenant.

/v3/users

HTTP Method

POST

Request Parameters

None

JSON Parameters

```
*username* - string value including network username for user (required)
*displayName* - string value displaying user name (required)
*email* - string value indicating email address for user (required)
*timeZone* - string value indicating location for user (required)
*enabled* - boolean value indicating user status (required)
    *true* - string value indicating enabled (required)
    *false* - string value indicating disabled user (required)
*orgRoleId* - integer value indicating role for user within organization (required)
    *1* - org_super_user
    *2* - org_full_user
    *3* - org_basic_user
    *7* - org_admin_only
*internalUserAvatarUrl* - string value indicating location of avatar image file for internal
    use
*userAvatarUrl* - string value indicating location of avatar image file
*languageId* - integer value indicating language used by user
```

Request Parameter Example

```
POST /v3/users
```

Sample Request

```
POST /v3/users
{
  "username": "Matt.Jones@example.com",
  "displayName": "Matt Jones",
  "email": "Matt.Jones@example.com",
  "timeZone": "Europe/London",
  "enabled": true,
  "avatarURL":
"http://a0.twimg.com/profile_images/1284202467/laptop-vector-model-122171296611479Nq0_normal.png",

  "languageId": 1,
  "internalUserAvatarUrl":
"http://socialstudio.s3.amazonaws.com/avatars/ad816b59936d93dc3ec36a43e15cbb8c"
}
```

Sample Response

A successful call returns a 201 code, indicating a successful creation.

A internal server error occuring during the call returns a 500 code. Try your call again later.

Update User

This method retrieves information for a specified user within a Social Studio tenant.

```
/v3/users/{id}
```

HTTP Method

```
PUT
```

Request Parameters

id - integer value indicating unique ID for user (you may also use the ***me*** shorthand in place of this value)

JSON Parameters

```
*username* - string value including network username for user (required)
*displayName* - string value displaying user name (required)
*email* - string value indicating email address for user (required)
*timeZone* - string value indicating location for user (required)
*enabled* - boolean value indicating user status (required)
    *true* - string value indicating enabled (required)
    *false* - string value indicating disabled user (required)
*orgRoleId* - integer value indicating role for user within organization (required)
    *1* - org_super_user
    *2* - org_full_user
    *3* - org_basic_user
    *7* - org_admin_only
*internalUserAvatarUrl* - string value indicating location of avatar image file for internal use
*userAvatarUrl* - string value indicating location of avatar image file
*languageId* - integer value indicating language used by user
```

Request Parameter Example

```
PUT /v3/users/{id}
```

Sample Request

```
PUT /v3/users/{id}
{
  "username": "Matt.Jones@example.com",
  "displayName": "Matt Jones",
  "email": "Matt.Jones@example.com",
  "timeZone": "Europe/London",
```

```
"enabled": true,  
"avatarURL":  
"http://a0.twimg.com/profile_images/1284202467/laptop-vector-model-122171296611479Nq0_normal.png",  
  
"languageId": 1,  
"internalUserAvatarUrl":  
"http://socialstudio.s3.amazonaws.com/avatars/ad816b59936d93dc3ec36a43e15cbb8c"  
}
```

Sample Response

A successful call returns a 204 No Content code, indicating a successful update.

A call referencing an invalid ID returns a 404 Not Found code. Ensure your call uses the correct ID.

A internal server error occuring during the call returns a 500 code. Try your call again later.

Reset Password

This method resets the password for an internal user account. A successful call also sends an email message (from support@radian6.com) containing a link the user can click to set a new password value. This call requires administrative privileges.

```
/v3/users/{userId}/password?action=reset
```

HTTP Method

```
POST
```

Request Parameters

```
*userId* - integer value indicating unique ID for user (also accepts me shorthand)
```

Request Parameters Example

```
POST /v3/users/{id}/password?action=reset
```

Sample Request

```
POST /v3/users/12345/password?action=reset
```

Sample Response

A successful call returns a 204 No Content code, indicating a successful update.

A call referencing an invalid ID returns a 404 Not Found code. Ensure your call uses the correct ID.

A internal server error occurring during the call returns a 500 code. Try your call again later.

Modify Password

This method triggers a password reset operation for a user

```
/users/{id}/password
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Sample Request

```
POST /users/1/password?action=change
```

CHAPTER 8 Workspaces

In this chapter ...

- [Retrieve Workspaces](#)
- [Retrieve Workspace](#)
- [Create Workspace](#)
- [Update Workspace](#)
- [Delete Workspace](#)
- [Create Workspace Resources](#)
- [Delete Workspace Resources](#)

Workspaces collect resources to support multiple social teams across a brand or regions. Workspaces also support collaboration with agencies while controlling application access. This section helps you create a workspace and add resources.

Get Workspaces

Before you create a new workspace, perform a retrieve on your existing workspaces to ensure one does not already exist that would fit your needs:

```
GET /v1/workspaces
```

cURL Example

```
curl -X GET -H "access_token: ce5c9c9f-470b-4efc-8254-db8649ed77bd"
-H "Content-Type: application/json"
'https://api.socialstudio.radian6.com/v1/workspaces'
```

Create Workspaces

Add a new workspace using the following call:

```
POST /v1/workspaces
```

cURL Examples

```
curl -X POST -H "access_token: ce5c9c9f-470b-4efc-8254-db8649ed77bd"
-H "Content-Type: application/json" -d '{
  "name": "Workspace Name",
  "description": "This workspace is for Campaign X",
  "logo": "/Pictures/workspace logo.png"
}' 'https://api.socialstudio.radian6.com/v1/workspaces'
```

Add a User to the Workspace

Add a user with the following call:

```
POST /v1/workspaceresources
```

cURL Examples

```
curl -X POST -H "access_token: your_access_token" -H "Content-Type: application/json" -d '{
  "workspace_id": "b8e4c9c4-9e3c-411f-ae80-7c2b3d6e2ab0",
  "resource_type": "user",
  "resource_id": "79799125",
  "access": "contributor"
}' 'https://api.socialstudio.radian6.com/v1/workspaces'
```

Retrieve Workspaces

This method retrieves a paginated list of workspaces within the Social Studio organization.

```
/v1/workspaces
```

HTTP Method

```
GET
```

Request Parameters

```
*page* - integer value indicating which page to display from retrieved data (default of 1)
*perpage* - integer value indicating how many results to include in a page (value must be greater than 1 and not exceed 25, default of 10)
*search* - string value indicating which characters to search for at start of workspace name (only at beginning, not within name strings)
*orderBy* - string value indicating order of returned results:
    *name* - name of workspace (default)
    *updated* - last modified timestamp
*order* - string value indicating display direction for retrieved content:
    *desc* - descending order
    *asc* - ascending order (default)
*user* - string value identifying workspace user. This parameter defaults to *default* value. The call ignores other values and returns an error for non-admin users.
*isMember* - boolean value that pairs with *user* parameter to return on workspaces with an assigned role for the user.
*visibility* - string value indicating whether a workspace appears to all users:
    *private* - does not appear (default)
    *public* - does appear
*include* - comma-separated string values of additional information to include:
    *permissions* - string value containing user permissions for the workspace
    *defaultPublishProfile* - string value of default publish profile for workspace
        *view* - can view social assets and posts
        *edit_content* - same as *view* and can create content (such as posts)
        *admin* - same as *edit_content* and can perform admin actions on workspace
        *delete* - can delete workspace
    *membercount* - string value containing total number of users for workspace
    *socialassetcount* - string value containing total number of social assets for workspace
```

Request Parameter Example

```
GET /v1/workspaces
```

Sample Request

The call below retrieves the workspaces within an account:

```
GET /v1/workspaces?visibility=public&search=salesforce&include=permissions,membercount
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": [
    {
      "name": "workspace A",
      "id": "46e66860-25fe-11e3-b1a1-168170973bd5",
      "visibility": "public",
      "logo": "a_logo.png",
      "description": "A great workspace",
      "group_id": "123"
    },
    {
      "name": "Workspace B",
      "id": "46f96860-25fe-11e3-b1a1-168170973bd5",
      "visibility": "public",
      "logo": "b_logo.png",
      "description": "B successful workspace",
      "group_id": "234"
    }
  ],
  "meta": {
    "total": 400,
    "page": 2,
    "perpage": 2,
    "current": "{HOST}/v1/workspaces?page=2&perpage=2&visibility=public",
    "prev": "{HOST}/v1/workspaces?page=1&perpage=2&visibility=public",
    "next": "{HOST}/v1/workspaces?page=3&perpage=2&visibility=public"
  }
}
```

Error Responses

A call finding no workspaces returns the following response:

```
{
  "status": true,
  "response": [],
  "meta": {
    "total": 0,
    "page": 1,
    "perpage": 10,
  }
}
```

```

    "current": "{HOST}/v1/workspaces?page=1&perpage=10&visibility=public",
    "prev": "",
    "next": ""
  }
}

```

A call including a bad argument returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": ""
      }
    ]
  },
  "meta": []
}

```

status - string value indicating success of call
 response - array of information indicating retrieval of workspaces
 name - string value including name of workspace
 id - string value of retrieved workspace
 visibility - string value indicating public or private workspace
 logo - string value including URL of logo image file
 description - string value including description of workspace
 group_id - reserved for internal use

Retrieve Workspace

This method retrieves a single workspace within the Social Studio organization.

```
/v1/workspaces/{id}
```

HTTP Method

```
GET
```

Request Parameters

id - string value identifying the workspace to retrieve

Request Parameter Example

```
GET /v1/workspaces/{id}
```


Query Parameters

```
*include* - comma-separated string values of additional information to include:
  *permissions* - string value containing user permissions for the workspace
  *defaultPublishProfile* - string value of default publish profile for workspace
    *view* - can view social assets and posts
    *edit_content* - same as *view* and can create content (such as posts)
    *admin* - same as *edit_content* and can perform admin actions on workspace
    *delete* - can delete workspace
  *membercount* - integer value including number of workspace members
  *socialassetcount* - integer value including number of workspace members
  *engage* - string value containing user engage permissions
```

Sample Request

The call below retrieves the specified workplace:

```
GET /v1/workspaces/46e66860-25fe-11e3-b1a1-168170973bd5?include=permissions,membercount
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "name": "A new workspace for the system",
    "id": "46e66860-25fe-11e3-b1a1-168170973bd5",
    "visibility": "private",
    "logo": "workspace_logo.png",
    "description": "A brand new workspace!",
    "group_id": "123"
  },
  "meta": []
}
```

A call returning information on a private workspace the user cannot access returns the following response:

```
{
  "status": true,
  "response": {
    "name": "PRIVATE WORKSPACE",
    "id": "46e66860-25fe-11e3-b1a1-168170973bd5",
    "visibility": "private"
  },
  "meta": []
}
```

A call including additional information returns the following response:

```
{
  "status": true,
  "response": {
```

```
{
  "id": "46e66860-25fe-11e3-b1a1-168170973bd5",
  "visibility": "public",
  "name": "A new workspace for the system",
  "logo": "",
  "description": "",
  "group_id": "123",
  "permissions": [
    "view",
    "edit_content",
    "admin",
    "delete"
  ],
  "membercount": 100,
  "socialassetcount": 50,
  "engage": [
    "view",
    "edit"
  ]
},
"meta": []
}
```

Error Responses

A call referencing an invalid workspace returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not_found",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

A call including an invalid ID returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": ""
      }
    ]
  },
}
```

```

    "meta": []
}

```

```

{
  "status": true,
  "response": {
    "id": "46e66860-25fe-11e3-b1a1-168170973bd5",
    "visibility": "public",
    "name": "A new workspace for the system",
    "logo": "",
    "description": "",
    "group_id": "123",
    "permissions": [
      "view",
      "edit_content",
      "admin",
      "delete"
    ],
    "membercount": 100,
    "socialassetcount": 50,
    "engage": [
      "view",
      "edit"
    ]
  },
  "meta": []
}

```

status - string value indicating success of call

response - array of information regarding retrieval of workspace

 id - string value of retrieved workspace

 visibility - string value indicating public or private workspace

 name - string value including name of workspace

 logo - string value including URL of logo image file

 description - string value including description of workspace

 group_id - string value including ID of associated group

 permissions - array of string values indicating permissions associated with workspace

 view

 edit_content

 admin

 delete

 membercount - integer value including number of workspace members

 socialassetcount - integer value including number of workspace members

 engage - array of actions associated with engage functions in workspace

 view

 *"edit"

Create Workspace

This method creates a workspace within a Social Studio organization.

```
/v1/workspaces
```

HTTP Method

```
POST
```

Request Parameters

```
*name* - string value containing name of new workspace (required). This value includes a
maximum of 255 characters. The call removes leading and trailing whitespaces. The call
replaces multiple consecutive whitespaces with a single whitespace.
*visibility* - string value indicating whether a workspace appears to all users:
    *private* - does not appear (default)
    *public* - does appear
*description* - string value containing text describing the new workspace (defaults to
empty string)
*logo* - string value indicating the location of a workspace image file (defaults to empty
string)
```

Request Parameter Example

```
POST /v1/workspaces
```

Sample Request

The call below creates a new piece of shared content with the specified message:

```
POST -d "name=A new workspace for the system" -d "visibility=private" -d
"logo=workspace_logo.png" -d "description=A brand new workspace." {HOST}/v1/workspaces
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "id": "95b75fac-0453-11e3-8d6f-168170973bd5"
  },
  "meta": []
}
```

Error Responses

An unsuccessful call due to a duplicate name returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.conflict",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

An unsuccessful call due to a bad parameter returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

status - string value indicating success of call
response - string value indicating creation of workspace
id - string value indicating ID of newly created workspace

Update Workspace

This method updates a workspace within a Social Studio organization.

```
/v1/workspaces
```

HTTP Method

```
PATCH
```

Request Parameters

id - string value containing the ID value for the workspace (required)

Request Parameter Example

```
PATCH /v1/workspaces/{id}
```

Query Parameters

```
*name* - string value containing name of new workspace. This value includes a maximum of
255 characters. The call removes leading and trailing whitespaces. The call replaces
multiple consecutive whitespaces with a single whitespace.
*visibility* - string value indicating whether a workspace appears to all users:
    *private* - does not appear (default)
    *public* - does appear
*description* - string value containing text describing the new workspace (defaults to
empty string)
*logo* - string value indicating the location of a workspace image file (defaults to empty
string)
```

Sample Request

The call below creates a new piece of shared content with the specified message:

```
PATCH -d "visibility=private" {HOST}/v1/workspaces/46e66860-25fe-11e3-b1a1-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": true,
  "meta": []
}
```

Error Responses

An call referencing a nonexistent workspace returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not.found",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

An unsuccessful call due to insufficient permissions returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

status - string value indicating success of call

response - string value indicating updating of workspace

Delete Workspace

This method deletes a workspace within the Social Studio organization. This process results in the loss of any scheduled posts.

```
/v1/workspaces/{id}
```

HTTP Method

```
DELETE
```

Request Parameters

id - string value containing the ID value for the workspace

Request Parameter Example

```
DELETE /v1/workspaces/{id}
```

Sample Request

The call below deletes the workspace and all associated content.

```
DELETE {HOST}/v1/workspaces/46e66860-25fe-11e3-b1a1-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": true,
  "meta": []
}
```

Error Responses

An call referencing a nonexistent workspace returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not_found",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

An unsuccessful call due to a bad parameter returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

An unsuccessful call due to insufficient permissions returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": ""
      }
    ]
  },
}
```



```

    "meta": []
  }

```

status - string value indicating success of call
response - string value indicating deletion of workspace

Create Workspace Resources

This method adds an existing resource to a workspace

```
/v1/workspaceresources
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Request Parameter Example

```
POST /v1/workspaceresources
```

JSON Parameters

Each JSON payload includes the resource type:

```

*workspace_id* - string value identifying owner of workspace resources (required)
*resource_type* - string value indicating type of workspace resource created by the call:

    *topic_profile*
    *engage_macro*
    *template*
    *social_property*
    *user*
    *publish_macro*
*resource_id* - string value containing identifier for the workspace resource
*access* - string value indicating user's workspace permission (required with the user
resource_type)
    *limited_member*
    *contributor*
    *admin*

```

Sample Request

The call below creates a new piece of shared content with the specified message:

```
POST    /v1/workspaceresources
{
  "workspace_id": "44dfccf0-ed00-4988-a256-0913281dcb89",
  "resource_type": "topic_profile",
  "resource_id": "19212"
}
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "resource_type": "topic_profile",
    "resource_id": "19212",
    "workspace_group_id": "8152"
  },
  "meta": []
}
```

status - string value indicating success of call
response - string value indicating creation of workspace resource
 resource_type - string value indicating type of workspace resource created by the call
 resource_id - string value containing identifier for the workspace resource
 workspace_group_id - reserved for internal use

Delete Workspace Resources

This method removes a resource from a workspace. This process does not delete the resource from the Social Studio organization.

```
/v1/workspaceresources
```

HTTP Method

```
DELETE
```

Request Parameters

```
None
```

Request Parameter Example

```
DELETE /v1/workspaceresources
```

JSON Parameters

Each JSON payload includes information on the shared content:

```
*workspace_id* - string value identifying owner of workspace resources (required).
*resource_type* - string value indicating type of workspace resource created by the call.
*resource_id* - string value containing identifier for the workspace resource.
```

Sample Request

The call below creates a new piece of shared content with the specified message:

```
DELETE /v1/workspaceresources
{
  "workspace_id": "44dfccf0-ed00-4988-a256-0913281dcb89",
  "resource_type": "topic_profile",
  "resource_id": "19212"
}
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "resource_type": "topic_profile",
    "resource_id": "19212",
    "workspace_group_id": "8152"
  },
  "meta": []
}
```

```
*status* - string value indicating success of call
*response* - string value indicating deletion of workspace resource
  *resource_type* - string value indicating type of workspace resource created by the call
  *resource_id* - string value containing identifier for the workspace resource
  *workspace_group_id* - reserved for internal use
```

CHAPTER 9 Imported Posts

In this chapter ...

- [Retrieve Posts & Metrics](#)

Use this endpoint to retrieve all brand posts and associated metrics. These posts could include any posts created within Social Studio or imported to Social Studio after posting on the social network itself. Make sure to include the specific metric parameters you want returned for the posts.

Retrieve the Number of Comments on Posts

Retrieve the number of comments on a post using the following call:

```
POST /v1/importedposts
```

cURL Example

```
curl -X POST -H "access_token: your_access_token" -H "Content-Type: application/json"
```

```
"https://api.socialstudio.radian6.com/v1/importedposts?startDate=1136500800&endDate=1437105599&metrics=comments"
```

Retrieve Posts & Metrics

This method retrieves posts and their metrics imported to a workspace.

```
/v1/importedposts
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*workspace* - UUID value indicating the workgroup (required)
*startDate* - date value in epoch indicating the start date from which to retrieve posts
(required)
*endDate* - date value in epoch indicating the end date from which to retrieve posts
(required)
*network* - string value indicating social network for post:
    *all* (default)
    *facebook*
    *twitter*
    *linkedin*
    *gplus*
    *youtube*
    *instagram*
*socialProperty* - string values (in a comma-delimited list) indicating the social properties
from which to retrieve the post (defaults to all social properties included in workspace)
*socialAsset* - string value indicating the Social Account Set ID from a workspace (defaults
to all social accounts included in workspace)
*metrics* - string values (in a comma-delimited list) included in the response:
    *facebook*
        *likes*
        *comments*
        *shares*
        *reach*
        *organic_reach*
        *viral_reach*
        *paid_reach*
        *fan_reach*
        *non_fan_reach*
        *fan_paid_reach*
        *engagement*
        *clicks*
        *engagement_rate*
    *twitter*
```

```

    *retweets*
    *replies*
    *favorites*
    *link_clicks*
    *engagement*
    *engagement_rate*
*linkedin*
    *likes*
    *comments*
    *link_clicks*
    *engagement*
*gplus*
    *plusone*
    *reshares*
    *comments*
    *link_clicks*
    *engagement*
*youtube*
    *views*
    *likes*
    *dislikes*
    *comments*
    *link_clicks*
    *engagement*
*instagram*
    *likes*
    *comments*
*pinterest*
    *likes*
    *comments*
    *repins*
    *link_clicks*
*labels* - string value (in a comma-delimited list) indicating labels to use in filter
*type* - string value indicating media type to use in filter
*orderBy* - string value indicating parameter used to sort results (defaults to date)
*order* - string value indicating display direction for retrieved content (defaults to
DESC):
    *DESC* - descending order (default)
    *ASC* - ascending order
*page* - integer value indicating which page to display from retrieved data (defaults to
1)
*perpage* - integer value indicating how many results to include in a page (defaults to
10)
*filter* - string value indicating publish status of post (defaults to unpublished)

```

Request Parameter Example

```
GET /v1/importedposts
```

Sample Request

The call below retrieves all imported posts from a specified workspace:

```
GET /v1/importedposts?workspace=1
```

The call below retrieves all imported Facebook posts from a specified workspace:

```
GET /v1/importedposts?workspace=1&network=facebook
```

The call below retrieves all imported Facebook posts from a specified workspace with specified metrics:

```
GET /v1/importedposts?workspace=1&network=facebook&metrics=likes,comments
```

The call below retrieves all imported Facebook posts from a specified workspace with specified metrics:

```
GET /v1/importedposts?workspace=1&network=facebook&metrics=likes,comments&orderBy=comments
```

Sample Response

Each imported post includes the following properties:

```
*id* - string value indicating ID of imported content
*socialAsset* - string value indicating ID of social asset
*socialProperty* - string value indicating ID of social property
*slug* - string value of slug for imported content
*assets* - array of information on assets within imported content
  *id* - numeric value including ID for imported asset
  *created* - timestamp value indicating asset creation data, in Unix time
  *metadata* - string value including metadata associated with imported asset
  *post_id* - string value including ID for post associated with asset
  *type* - integer value indicating type of asset
    *1* - Album
    *2* - Image
    *3* - Link
    *4* - Question
    *5* - Video
  *updated* - timestamp value indicating asset updating data, in Unix time
  *url* - string value including URL for asset
*date* - timestamp value indicating date for content creation, in Unix time
*message* - string value including message of content
*network* - string value including social network used for content
*targeting* - string value indicating targeting for content
*labels* - string value indicating labels for content
*type* - string value indicating type of content within social network (such as update)
*socialAssetName* - string value including name of content from social network
*blogPostId* - string value indicating ID of associated blog post
*metrics* - array of information about social content analytics
*likes* - integer value of social network likes
*comments* - integer value of social network comments
```

A successful call returns posts from all networks in last seven days in descending order:

```
{
  "status": true,
```

```

"response": [
  {
    "id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
    "socialAsset": "200012913364042",
    "socialProperty": 8307,
    "slug": "/Buddy-The-Dog/200012913364042",
    "assets": {
      "id": 46,
      "created": "2013-08-12T10:48:28-0400",
      "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing  
To Grow Your Business -  
Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change  
stuff  
ne\",\"tags\":\"[http://www.salesforce.com/using/your-way]\",\"tag_index\":0,\"thumbnail\":\"\",\"link\":\"https://salesforce.com\",\"video\":\"nil\",\"type\":\"Link\"\",
      "post_id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
      "type": 3,
      "updated": "2013-08-12T10:48:28-0400",
      "url": "https://salesforce.com"
    },
    "date": "2013-08-13 09:45:03",
    "message": "Salesforce.com is cool!",
    "network": "facebook",
    "targeting": [ ],
    "labels": [ ],
    "type": "status",
    "socialAssetName": "Buddy - The Dog",
    "blogPostId": 1234
  },
  {
    "id": "b93ab12e-37cd-4344-bffa-3a59f5053580",
    "socialAsset": "200012913364042",
    "socialProperty": 8307,
    "slug": "/Buddy-The-Dog/200012913364042",
    "assets": [ ],
    "date": "2013-08-12 15:29:40",
    "message": "yahoo.com",
    "network": "facebook",
    "targeting": [ ],
    "labels": [ ],
    "type": "status",
    "socialAssetName": "Buddy - The Dog",
    "blogPostId": 1234
  },
  {
    "id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
    "socialAsset": "200012913364042",
    "socialProperty": 8307,
    "slug": "/Buddy-The-Dog/200012913364042",
    "assets": {
      "id": 47,
      "created": "2013-08-12T10:48:29-0400",
      "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing  
To Grow Your Business -

```



```

Salesforce.com\", \"caption\": \"https://salesforce.com\", \"description\": \"You can change
stuff
ae\", \"tags\": [\"http://static.com/assets/logo.png\", \"facebook\", \"thumbnail\", \"socialLink\", \"file\", \"id\": \"https://salesforce.com\", \"video\": null, \"type\": \"link\"],

    \"post_id\": \"9bc25322-49ca-4b9c-ae81-20ff1b593edf\",
    \"type\": 3,
    \"updated\": \"2013-08-12T10:48:29-0400\",
    \"url\": \"https://salesforce.com\"
  },
  \"date\": \"2013-08-12 09:45:03\",
  \"message\": \"Salesforce.com is cool!\",
  \"network\": \"facebook\",
  \"targeting\": [ ],
  \"labels\" : [ ],
  \"type\": \"status\",
  \"socialAssetName\": \"Buddy - The Dog\",
  \"blogPostId\": 1234
}
...
],
\"meta\": {
  \"total\": 6,
  \"perpage\": 10,
  \"current\": 1,
  \"prev\": \"\",
  \"next\": \"\"
}
}

```

A successful call returns posts from Facebook in last seven days in descending order:

```

{
  \"status\": true,
  \"response\": [
    {
      \"id\": \"35db3b27-edd5-4d35-8ef7-574b0f483524\",
      \"socialAsset\": \"200012913364042\",
      \"socialProperty\": 8307,
      \"slug\": \"/Buddy-The-Dog/200012913364042\",
      \"assets\": {
        \"id\": 46,
        \"created\": \"2013-08-12T10:48:28-0400\",
        \"metadata\": \"{\\\"link_type\\\":\\\"link\\\",\\\"title\\\":\\\"CRM and Cloud Computing
To Grow Your Business -
Salesforce.com\\\",\\\"caption\\\":\\\"https://salesforce.com\\\",\\\"description\\\":\\\"You can change
stuff
ae\", \"tags\": [\"http://static.com/assets/logo.png\", \"facebook\", \"thumbnail\", \"socialLink\", \"file\", \"id\": \"https://salesforce.com\", \"video\": null, \"type\": \"link\"],

        \"post_id\": \"35db3b27-edd5-4d35-8ef7-574b0f483524\",
        \"type\": 3,
        \"updated\": \"2013-08-12T10:48:28-0400\",
        \"url\": \"https://salesforce.com\"
      },
      \"date\": \"2013-08-13 09:45:03\",
      \"message\": \"Salesforce.com is cool!\",
    }
  ]
}

```

```

        "network": "facebook",
        "targeting": [ ],
        "labels" : [ ],
        "type": "status",
        "socialAssetName": "Buddy - The Dog",
        "blogPostId": 1234
    },
    {
        "id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
        "socialAsset": "200012913364042",
        "socialProperty": 8307,
        "slug": "/Buddy-The-Dog/200012913364042",
        "assets": {
            "id": 47,
            "created": "2013-08-12 10:48:29",
            "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing To Grow Your Business - Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change stuff here\",\"tags\":{\"http://www.salesforce.com/assets/logo.png\"},\"is_image\":true,\"is_thumbnail\":true,\"is_link\":false,\"id\":\"https://salesforce.com\",\"video\":null,\"type\":\"link\"\",
            "post_id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
            "type": 3,
            "updated": "2013-08-12T10:48:29-0400",
            "url": "https://salesforce.com"
        },
        "date": "2013-08-12 09:45:03",
        "message": "Salesforce.com is cool!",
        "network": "facebook",
        "targeting": [ ],
        "labels" : [ ],
        "type": "status",
        "socialAssetName": "Buddy - The Dog",
        "blogPostId": 1234
    }
    ...
],
"meta": {
    "total": 16,
    "perpage": 10,
    "current": 1,
    "prev": "",
    "next": ""
}
}

```

A successful call returns posts from Facebook in last seven days in descending order with metrics:

```

{
    "status": true,
    "response": [
        {
            "id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
            "socialAsset": "200012913364042",
            "socialProperty": 8307,

```

```

"slug": "/Buddy-The-Dog/200012913364042",
"assets": {
  "id": 46,
  "created": "2013-08-12T10:48:28-0400",
  "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing To Grow Your Business - Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change stuff\"}",
  "url": "https://salesforce.com/assets/info-comp.jpg", "caption": "info-comp", "thumbnail": "https://salesforce.com/assets/info-comp.jpg", "link": "https://salesforce.com/assets/info-comp.jpg", "type": "link", "video": null, "type": "link"},
  "post_id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
  "type": 3,
  "updated": "2013-08-12T10:48:28-0400",
  "url": "https://salesforce.com"
},
{
  "date": "2013-08-13 09:45:03",
  "message": "Salesforce.com is cool!",
  "network": "facebook",
  "targeting": [ ],
  "labels" : [ ],
  "type": "status",
  "socialAssetName": "Buddy - The Dog",
  "blogPostId": 1234,
  "metrics": {
    "likes": 300,
    "comments": 400
  }
},
{
  "id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
  "socialAsset": "200012913364042",
  "socialProperty": 8307,
  "slug": "/Buddy-The-Dog/200012913364042",
  "assets": {
    "id": 47,
    "created": "2013-08-12T10:48:29-0400",
    "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing To Grow Your Business - Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change stuff\"}",
    "url": "https://salesforce.com/assets/info-comp.jpg", "caption": "info-comp", "thumbnail": "https://salesforce.com/assets/info-comp.jpg", "link": "https://salesforce.com/assets/info-comp.jpg", "type": "link", "video": null, "type": "link"},
    "post_id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
    "type": 3,
    "updated": "2013-08-12T10:48:29-0400",
    "url": "https://salesforce.com"
  },
  "date": "2013-08-12 09:45:03",
  "message": "Salesforce.com is cool!",
  "network": "facebook",
  "targeting": [ ],
  "labels" : [ ],
  "type": "status",
  "socialAssetName": "Buddy - The Dog",

```

```

        "blogPostId": 1234,
        "metrics": {
            "likes": 100,
            "comments": 800
        }
    },
    ...
],
"meta": {
    "total": 16,
    "perpage": 10,
    "current": 1,
    "prev": "",
    "next": ""
}
}

```

A successful call returns posts from Facebook in last seven days sorted by comments in descending order with metrics:

```

{
    "status": true,
    "response": [
        {
            "id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
            "socialAsset": "200012913364042",
            "socialProperty": 8307,
            "slug": "/Buddy-The-Dog/200012913364042",
            "assets": {
                "id": 46,
                "created": "2013-08-12T10:48:28-0400",
                "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing To Grow Your Business - Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change stuff here\",\"tags\":\"http://www.salesforce.com/salesforce.com\",\"tag_id\":\"\",\"thumbnail\":\"\",\"src_link\":\"https://salesforce.com\",\"video\":\"\",\"type\":\"link\"\",
                "post_id": "35db3b27-edd5-4d35-8ef7-574b0f483524",
                "type": 3,
                "updated": "2013-08-12T10:48:28-0400",
                "url": "https://salesforce.com"
            },
            "date": "2013-08-13 09:45:03",
            "message": "Salesforce.com is cool!",
            "network": "facebook",
            "targeting": [ ],
            "labels": [ ],
            "type": "status",
            "socialAssetName": "Buddy - The Dog",
            "blogPostId": 1234,
            "metrics": {
                "likes": 300,
                "comments": 800
            },
            "blogPostId": 1234
        },
    ],
}

```

```

    {
      "id": "9bc25322-49ca-4b9c-ae81-20ff1b593edf",
      "socialAsset": "200012913364042",
      "socialProperty": 8307,
      "slug": "/Buddy-The-Dog/200012913364042",
      "assets": {
        "id": 47,
        "created": "2013-08-12T10:48:29-0400",
        "metadata": "{\"link_type\":\"link\",\"title\":\"CRM and Cloud Computing  
To Grow Your Business -  
Salesforce.com\",\"caption\":\"https://salesforce.com\",\"description\":\"You can change  
stuff  
ne\",\"tags\":[\"http://www.static.com/assets/logo.png\"],\"tag_ids\":[1],\"thumbal\":\"true\",\"base_link\":\"file\",\"id\":\"https://salesforce.com\",\"video_c\":\"nil\",\"type\":\"Link\"\",
        \"post_id\": \"9bc25322-49ca-4b9c-ae81-20ff1b593edf\",
        \"type\": 3,
        \"updated\": \"2013-08-12T10:48:29-0400\",
        \"url\": \"https://salesforce.com\"
      },
      \"date\": \"2013-08-12 09:45:03\",
      \"message\": \"Salesforce.com is cool!\",
      \"network\": \"facebook\",
      \"targeting\": [ ],
      \"labels\" : [ ],
      \"type\": \"status\",
      \"socialAssetName\": \"Buddy - The Dog\",
      \"blogPostId\": 1234,
      \"metrics\": {
        \"likes\": 100,
        \"comments\": 600
      }
    }
  ],
  \"meta\": {
    \"total\": 16,
    \"perpage\": 10,
    \"current\": 1,
    \"prev\": \"\",
    \"next\": \"\"
  }
}

```

CHAPTER 10 Shared Content

In this chapter ...

- [Retrieve Shared Content](#)
- [Share & Unshare Content](#)
- [Create Shared Content](#)
- [Archive & Unarchive Shared Content](#)
- [Delete Shared Content](#)

Share content across teams using this functionality. Agencies can also share content after finishing the planning phase for your content. Create the content from an existing post, then share it to other workspaces or send it to another tool (such as a content management system).

Create Shared Content

Ensure you host the image before sharing it:

```
POST /v1/clips
```

cURL Example

```
curl -X POST -H "access_token: a133c435-dd3a-4727-8c07-08e09f0b1a97"
-H "Content-Type: application/json" -d '{
  "message": "Message text",
  "description": "Optional description about why I am sharing this
content to others",
  "assets": [{"url": "http://www.foo.com/images/foo.png", "type": 2,
"metadata": ""}],
  "workspaces": ["88c282e3-b33e-4109-bb86-9040c368bf75"],
  "workspace_id": "b8e4c9c4-9e3c-411f-ae80-7c2b3d6e2ab0"
}' 'https://api.socialstudio.radian6.com/v1/clips'
```

Create Shared Content and Associate It to an Existing Post

Ensure you host the image before sharing it:

```
POST /v1/clips
```

cURL Example

```
curl -X POST -H "access_token: 67fbd2fa-c885-4c53-a383-9d4d18e1763b"
-H "Content-Type: application/json" -H -d '{
  "post_id": "65fe0314-9737-4fd3-acd5-a7c1cf636b7a",
  "message": "This duplicates that content from an existing post
```

```
into a new piece of shared content, and associates them together when
measuring performance of content.",
  "description": "This image did very well on Pinterest, please try
it on your regional accounts.",
  "assets": [{"url": "http://www.foo.com/images/foo.png", "type": 2,
"metadata": ""}],
  "workspaces": ["88c282e3-b33e-4109-bb86-9040c368bf75"],
  "workspace_id": "b8e4c9c4-9e3c-411f-ae80-7c2b3d6e2ab0"
}' 'https://api.socialstudio.radian6.com/v1/clips'
```

Delete Shared Content

This call removes content from all workspaces and removes access to any analytics for the content:

```
DELETE /v1/clips/{id}
```

cURL Example

```
curl -X DELETE -H "access_token: a133c435-dd3a-4727-8c07-08e09f0b1a97"
-H
'https://api.socialstudio.radian6.com/v1/clips/e15318fa-c194-45e1-99d3-05f39e103e0f'
```

Archive Shared Content

This call disables usage of shared content for a new post. However, users in workspaces to which you shared the content can still view performance metrics for the content:

```
POST /v1/clips/{id}?action=archive
```

cURL Example

```
curl -X POST -H "access_token: 67fbd2fa-c885-4c53-a383-9d4d18e1763b"
-H "Content-Type: application/json"
-d ''
'https://api.socialstudio.radian6.com/v1/clips/22a27159-832c-47f1-9313-5533dfe40a8d?action=archive'
```

Unshare Shared Content

To unshare content from one or more recipient workspaces, define an array of workspaces from which to unshare. Ensure you provide only workspaces where it has been shared to previously:

```
POST /v1/clips/{id}?action=unshare
```

cURL Example

```
curl -X POST -H "access_token: 67fbd2fa-c885-4c53-a383-9d4d18e1763b"
-H "Content-Type: application/json" -d '{
  "workspaces": ["a0fd4ce7-af4c-48b1-b6e8-0cb4ce54b2fc",
"c2e918d2-4309-11e3-bb55-1cb63b08732c"]
}'
'https://api.socialstudio.radian6.com/v1/clips/22a27159-832c-47f1-9313-5533dfe40a8d?action=unshare'
```


Retrieve Shared Content

This method retrieves content available for use within a workspace.

```
/v1/clips/
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*workspace* - string value indicating workspace from which to retrieve content (required)
*filter* - string value indicating which content to retrieve:
    *all* - any content (default)
    *owner* - any content owned by a specific owner
    *shared* - any content shared for use
*include* - string value requesting additional information:
    *metrics* - analytics information for specified content
*sort* - string value indicating order to display retrieved content:
    *created* - sort by created date (default)
    *avg_engagement* - sort by average number of engagement
*direction* - string value indicating display direction for retrieved content:
    *desc* - descending order (default)
    *asc* - ascending order
*page* - integer value indicating which page to display from retrieved data (defaults to 1)
*perpage* - integer value indicating how many results to include in a page (defaults to 10)
```

Request Parameter Example

```
GET /v1/clips
```

Sample Request

The call below retrieves all clips from the specified workplace with analytics data:

```
GET /v1/clips?workspace=584d3ad3-2f7b-11e3-b8e3-168170973bd5&filter=all&include=metrics
```

The call below retrieves a clip by ID:

```
GET v1/clips/35543aba-18b2-4995-bb76-ea1132fe544a
```

Sample Response

A successful call requesting all clips with analytics data returns the following response:

```
{
  "status": true,
  "response": [
    {
      "id": "35543aba-18b2-4995-bb76-ea1132fe544a",
      "source_type": "post",
      "source_id": "93023aba-18b2-4995-bb76-ea1132fe544a",
      "organization_id": "10567",
      "archived": 0,
      "deleted": 0,
      "message": "Post message",
      "description": "Clip comment",
      "start_date": null,
      "end_date": null,
      "created": "2014-02-02 11:00:00",
      "updated": "2014-02-03 11:00:00",
      "assets": [
        {
          "url": "https://d2ee3wvl22m8i7.cloudfront.net/_blue.jpg",
          "type": "2",
          "metadata": null
        }
      ],
      "owner_workspace": {
        "id": "75443f9c-349b-4bfa-a124-b188d4a8719f",
        "owner": "1",
        "name": "Buddy - The Dog",
        "logo": null
      },
      "shared_workspace_count": 0,
      "shared_workspaces": [
        {
          "id": "adlskasdaf-dsafada",
          "name": "workspace 1",
          "logo": "http://image.jpg"
        },
        {
          "id": "adlskasdaf-dsafada",
          "name": "workspace 2",
          "logo": "http://image.jpg"
        }
      ],
      "post_count": 19,
      "metrics": {
        "engagement": 354,
        "average_engagement": 177,
        "engagement_rate": {
          "facebook": 1.4,
          "twitter": 1.5
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    ...
  },
  ...
]
"meta": {
  "total": 5,
  "perpage": 10,
  "current": 1,
  "prev": "",
  "next": "/clips?page=2&perpage=10"
}
}

```

A successful call for a specific piece of content returns the following response:

```

{
  "status": true,
  "response": {
    "id": "35543aba-18b2-4995-bb76-ea1132fe544a",
    "source_type": "post",
    "source_id": "93023aba-18b2-4995-bb76-ea1132fe544a",
    "organization_id": "10567",
    "archived": 0,
    "deleted": 0,
    "message": "Post message",
    "description": "Clip comment",
    "start_date": null,
    "end_date": null,
    "created": "2014-02-02 11:00:00",
    "updated": "2014-02-03 11:00:00",
    "assets": [
      {
        "url": "https://d2ee3wvl22m8i7.cloudfront.net/_blue.jpg",
        "type": "2",
        "metadata": null
      }
    ],
    "owner_workspace": {
      "id": "75443f9c-349b-4bfa-a124-b188d4a8719f",
      "owner": "1",
      "name": "Buddy - The Dog",
      "logo": null
    },
    "shared_workspace_count": 0,
    "shared_workspaces": [
      {
        "id": "adlskasdaf-dsafada",
        "name": "workspace 1",
        "logo": "http://image.jpg"
      },
      {

```

```

        "id": "adlskasdaf-dsafada",
        "name": "workspace 2",
        "logo": "http://image.jpg"
    },
    "post_count": 19,
    "metrics": {
        "engagement": 354,
        "average_engagement": 177,
        "engagement_rate": {
            "facebook": 1.4,
            "twitter": 1.5
        }
    }
},
"meta": []
}

```

status - string value indicating success of call

response - array of information related to retrieved content:

- *id* - string value indicating ID of content
- *source_type* - string value indicating method of creation for content
- *source_id* - string value containing ID of content source
- *organization_id* - sting value containing ID of organization
- *archived* - integer value indicating number of archived posts
- *deleted* - integer value indicating number of deleted posts
- *message* - string value including content message
- *description* - string value including description of content
- *start_date* - timestamp value including start date for content
- *end* - timestamp value including end date for content
- *created* - timestamp value including creation date for content
- *updated* - timestamp value including update date for content
- *assets* - array value including information on content
 - *url* - string value including URL for content assets
 - *type* - integer value indicating the media type for the imported post:
 - *1* - MEDIA_TYPE_ALBUM
 - *2* - MEDIA_TYPE_LINK
 - *3* - MEDIA_TYPE_PHOTO
 - *4* - MEDIA_TYPE_STATUS
 - *5* - MEDIA_TYPE_VIDEO
- *metadata* - additional data for content
- *owner_workspace* - array value including data on workspacing owning content
 - *id* - string value including ID for owner workspace
 - *owner* - integer value indicating owner
 - *name* - string value including name of content in workspace
 - *logo* - string value indicating logo assoaited with content
- *shared_workspace_count* - integer value indicating number of workspaces sharing content
- *shared_workspaces* - array including information on workspaces sharing content
 - *id* - string value indicating ID of workspace sharing content
 - *name* - string value including name of workspace sharing content
 - *logo* - string value including location of logo image file
- *post_count* - integer value indicating number of times post used

```
*metrics* - array of information on content
*engagement* - integer value including number of engagements
*average_engagement* - integer value including average number of engagement
*engagement_rate* - array of engagement information by social network
  *facebook* - integer value including engagement rate on Facebook
  *twitter* - integer value including engagement rate on Twitter
```

Share & Unshare Content

This method shares and unshares content with other workspaces.

```
/v1/clips/{id}
```

HTTP Method

```
POST
```

Request Parameters

```
*id* - ID for the clip (required)
*action* - string value indicating the action to perform on the content:
  *share* - share content with other workspaces in the Social Studio organization
  *unshare* - unshare content with other workspaces in the Social Studio organization
```

Request Parameter Example

```
POST /v1/clips/{ID}?action=share
```

JSON Parameters

```
*workspaces* - array of string value indicating the workspaces with which to share the content
```

Sample Request

The call below shares the specified content:

```
POST v1/clips/35543aba-18b2-4995-bb76-ea1132fe544a?action=share
```

Sample Response

A successful call returns the following response:

```
# HTTP 200 Success
{
  "status": true,
  "response": {},
  "meta": {},
}
```

status - string value indicating success of call

Create Shared Content

This method creates shared content and allows for distribution to other workspaces.

```
/v1/clips/
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Request Parameter Example

```
POST /v1/clips
```

JSON Parameters

Each JSON payload includes information on the shared content:

```
*message* - string value indicating the message to share (required). The JSON payload must
include a *message* value or an *assets* value. You can include both.
*assets* - string value indicating URL and type of content (required). The JSON payload
must include a *message* value or an *assets* value. You can include both.
  *url* - string value indicating URL for content
  *type* - integer value indicating the media type for the imported content:
    *1* - MEDIA_TYPE_ALBUM
    *2* - MEDIA_TYPE_LINK
    *3* - MEDIA_TYPE_PHOTO
    *4* - MEDIA_TYPE_STATUS
    *5* - MEDIA_TYPE_VIDEO
*workspaces* - string value identifying the workspace in which you can access the content
```

```
(required).
*workspace_id* - string value identifying owner of shared content (required). You can leave
this value blank.
*description* - string value containing text describing the shared content.
*post_id* - string value containing the GUID for the existing post containing the shared
content to link and use for performance reporting
*clip_id* - string value containing a GUID for a clip containing the shared content to
link and use for performance reporting
```

Sample Request

The call below creates a new piece of shared content with the specified message:

```
{
  "message": "This is a test message",
  "workspaces": ["c2e918d2-4309-11e3-bb55-1cb63b08732d"],
  "workspace_id": "584d3ad3-2f7b-11e3-b8e3-168170973bd5"
}
```

The call below creates a new piece of shared content with the specified asset:

```
{
  "assets": [{"url": "http://www.foo.com", "type": 2}],
  "workspaces": ["c2e918d2-4309-11e3-bb55-1cb63b08732d"],
  "workspace_id": "584d3ad3-2f7b-11e3-b8e3-168170973bd5"
}
```

The call below creates a new piece of shared content and shares it to the specified workspaces:

```
{
  "message": "This is a test message",
  "workspace_id": "584d3ad3-2f7b-11e3-b8e3-168170973bd5",
  "description": "This is a test comment",
  "assets": [{"url": "http://www.foo.com", "type": 3,
"metadata": "{\\"url\\":\\"http://www.foo.com\\"}" }],
  "workspaces": ["c2e918d2-4309-11e3-bb55-1cb63b08732d"]
}
```

The call below creates a new piece of shared content containing an image from an existing post:

```
{
  "post_id": "82e6c061-2c5a-11e3-b1a1-168170973bd5"
  "message": "This is a test message",
  "workspace_id": "584d3ad3-2f7b-11e3-b8e3-168170973bd5",
  "description": "This is a test comment",
  "assets": [{"url": "http://www.foo.com/images/foo.png", "type": 2}],
  "workspaces": ["c2e918d2-4309-11e3-bb55-1cb63b08732d"]
}
```

The call below duplicate a piece of shared content containing an link from an existing clip:

```
{
  "clip_id": "82e6c061-2c5a-11e3-b1a1-168170973bd5"
  "message": "This is a test message",
  "workspace_id": "584d3ad3-2f7b-11e3-b8e3-168170973bd5",
  "description": "This is a test comment",
}
```

```
"assets": [{"url":"http://www.foo.com", "type":3}],
"workspaces": ["c2e918d2-4309-11e3-bb55-1cb63b08732d"]
}
```

Sample Response

A successful call returns the following response:

```
# HTTP 201 Created
{
  status: true,
  response: {
    id: "35543aba-18b2-4995-bb76-ea1132fe544a",
  },
  meta: {},
}
```

status - string value indicating success of call

id - string value indicating ID for newly created content

Archive & Unarchive Shared Content

This method archives and unarchives content within a workspace.

```
/v1/clips/{id}
```

HTTP Method

```
POST
```

Request Parameters

id - ID for the clip (required)

action - string value indicating the action to perform on the content:

 archive - archive the content, making it available for review but not for use

 unarchive - unarchive the content and make it available for use

Request Parameter Example

```
POST /v1/clips/{ID}?action=archive
```


Sample Request

The call below archives the specified content:

```
POST v1/clips/35543aba-18b2-4995-bb76-ea1132fe544a?action=archive
```

Sample Response

A successful call returns the following response:

```
# HTTP 200 Success
{
  "status": true,
  "response": {},
  "meta": {},
}
```

status - string value indicating success of call

Delete Shared Content

This method deletes content within the origin and recipient workspaces.

```
/v1/clips/{id}
```

HTTP Method

```
DELETE
```

Request Parameters

ID - ID for the clip (required)

Request Parameter Example

```
DELETE /v1/clips/{id}
```

Sample Request

The call below deletes the specified content:

```
DELETE v1/clips/35543aba-18b2-4995-bb76-ea1132fe544a
```

Sample Response

A successful call returns the following response:

```
# HTTP 200 Success
{
  "status": true,
  "response": {},
  "meta": {},
}
```

status - string value indicating success of call

CHAPTER 11 Short URLs

In this chapter ...

- [Retrieve Short URLs](#)
- [Retrieve Short URL](#)
- [Create Short URL](#)
- [Delete Short URL](#)

Short URLs allow brands to manage URL shortening services within the suite. Currently, Social Studio supports shortening and tracking clicks on links using the Bitly URL shortening service. Add your own Bitly accounts in order to shorten links using your own vanity URLs and track clicks both within Bitly as well as within Social Studio. Link tracking only applies to links published via Social Studio.

Add a Bitly Account

Add a Bitly account to a workspace using the following call:

```
POST /v1/shorturls
```

cURL Example

```
curl -X POST -H "access_token: 418adb62-b49f-46f7-be1e-c4d82c4b0fef"  
  
https://api.socialstudio.radian6.com/v1/shorturls?workspaceId=1649493c-411f-a80-723b62b0&accessToken={bitlyToken}'
```

Retrieve Short URLs

This method returns Bitly accounts within a workspace.

```
/v1/shorturls
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*workspace* - string value indicating unique ID for workspace (required)
*page* - integer value indicating number of result pages to return as part of the call
(defaults to 1)
*perpage* - integer value indicating number of results to return per page (range from 1
to 25, defaults to 10)
*search* - string value indicating the first characters in the titles as part of the search
(returns only characters from the start of the title)
```

Request Parameters Example

```
GET /v1/shorturls
```

Sample Request

```
curl GET
'{HOST}/v1/shorturls?workspace=028e2d63-0914-11e3-8d6f-168170973bd5&page=2&perpage=2'
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": [
    {
      "customShortDomain": "gofi.gs ",
      "id": "5239534f-0b58-11e3-8d6f-168170973bd5",
      "login": "dummy",
      "token": "thisisatestaccesstoken1"
    },
  ],
}
```

```

    {
      "customShortDomain": "gofi.gs ",
      "id": "9e9318c5-0c00-11e3-8441-168170973bd5",
      "login": "dummy",
      "token": "thisisatestaccesstoken2"
    }
  ],
  "meta": {
    "total": 4,
    "page": 1,
    "perpage": 2,
    "current":
"{HOST}/v1/shorturls?perpage=2&workspace=028e2d63-0914-11e3-8d6f-168170973bd5&page=1",
    "prev": "",
    "next":
"{HOST}/v1/shorturls?perpage=2&workspace=028e2d63-0914-11e3-8d6f-168170973bd5&page=2"
  }
}

```

response - array of objects containing information on
 customShortDomain - string value indicating Bitly account domain
 id - string value indicating unique ID for Bitly account within workspace
 login - string value indicating Bitly account user
 token - string value indicating OAuth user token

A call that returns no results based on the provided criteria includes the following response:

```

{
  "status": true,
  "response": [],
  "meta": {
    "total": 0,
    "page": 1,
    "perpage": 10,
    "current":
"{HOST}/v1/shorturls?workspace=14a6e8f7-2c59-11e3-b1a1-168170973bd5&page=1",
    "prev": "",
    "next": ""
  }
}

```

Modify your criteria as needed.

A call containing an invalid parameter returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "{{is dependent upon the bad argument}}"
      }
    ]
  },
}

```

```
    "meta": []  
  }
```

Ensure you enter the correct parameters for your call.

Retrieve Short URL

This method returns a specific Bitly account within a workspace.

```
/v1/shorturls/{id}
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - string value indicating unique ID for Bitly account
```

Request Parameters Example

```
GET /v1/shorturls{id}
```

Sample Request

```
curl GET {HOST}/v1/shorturls/123
```

Sample Response

A successful call returns the following response:

```
{  
  "status": true,  
  "response": [  
    {  
      "customShortDomain": "www.cus.tom",  
      "id": "5239534f-0b58-11e3-8d6f-168170973bd5",  
      "login": "dummy",  
      "token": "thisisatestaccesstoken3"  
    },  
  ],  
}
```

```
*response* - array of objects containing information on Bitly account  
  *customShortDomain* - string value indicating Bitly account domain
```

```
*id* - string value indicating unique ID for Bitly account within workspace
*login* - string value indicating Bitly account user
*token* - string value indicating OAuth user token
```

A call that returns no results based on the provided criteria includes the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not_found",
        "message": "Could not find a shorturl with that id"
      }
    ]
  },
  "meta": []
}
```

Ensure you use the correct ID value in your call.

A call containing an invalid parameter returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "{{is dependent upon the bad argument}}"
      }
    ]
  },
  "meta": []
}
```

Ensure you enter the correct parameters for your call.

Create Short URL

This method associates a Bitly account.

```
/v1/shorturls
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

JSON Parameters

```
*id* - string value including the unique identifier for the workspace
*accessToken* - string value containing the OAuth access token generated for accessing the
bit.ly account
```

Request Parameters Example

```
POST /v1/shorturls
```

Sample Request

```
curl -X POST -H "access_token: 1111-111-1111-11111111" -F
"accessToken=24847bd4936ac9667f1fb7e8c6fec2d5dc2db641" -F
"workspace=aa69ff7d-2b11-478a-9cc9-3fd3a5e1e19e" "{HOST}/v1/shorturls"
```

Sample Response

A successful call returns the following response:

```
{
  "status":true,
  "response":{
    "id":"5239534f-0b58-11e3-8d6f-168170973bd5"
  },
  "meta":[]
}
```

```
*id* - string value indicating unique ID for the URL shortener within the workspace
```

A call referencing an account already associated with a workspace returns the following response:

```
{
  "status":false,
  "response":{
    "errors":[
      {
        "code":"error.conflict",
        "message":"this login already exists in the workspace."
      }
    ]
  },
  "meta":[]
}
```

Ensure that you create a new account for the workspace or use an update call.

A call containing an invalid parameter returns the following response:

```
{
  "status":false,
  "response":{
    "errors":[
      {
        "code":"error.bad_request",
        "message":"Workspace not found"
      }
    ]
  },
  "meta":[]
}
```

Ensure you enter the correct parameters for your call.

Delete Short URL

This method deletes all Bitly accounts associated with a workspace.

```
/v1/shorturls
```

HTTP Method

```
DELETE
```

Request Parameters

id - string value indicating unique ID for Bitly account

Request Parameters Example

```
DELETE /v1/shorturls/{id}
```

Sample Request

```
curl -X DELETE {HOST}/v1/shorturls/740a25ea-0b57-11e3-8d6f-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  "status":true,
  "response":{
    "id":"5239534f-0b58-11e3-8d6f-168170973bd5"
  }
}
```

```
    },  
    "meta": []  
}
```

A call that returns no results based on the provided criteria includes the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.not_found",  
        "message": "Could not find a shorturl with that id"  
      }  
    ]  
  },  
  "meta": []  
}
```

Ensure you use the correct ID value in your call.

A call without the proper permissions returns the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.forbidden",  
        "message": "You do not have access to delete"  
      }  
    ]  
  },  
  "meta": []  
}
```

Ensure you enter the correct parameters for your call.

CHAPTER 12 Publish Macros

In this chapter ...

- [Retrieve Publish Macros](#)
- [Retrieve Publish Macro](#)
- [Retrieve Publish Macro Attribute](#)
- [Create Publish Macro](#)
- [Delete Publish Macro](#)

Publish Macros allow you to configure predefined metadata for several attributes for use (even automatically applied) when creating a brand post within a workspace.

Attributes refer to categories of metadata, such as web analytics, publish labels, or social accounts.

Creating a Publish Macro

This example includes creating a new publish macro with attributes for web analytics tracking codes in the Social Studio organization

```
POST /v3/publishprofiles
```

cURL Example

```
curl -X POST -H "access_token: a78cabb9-c00e-4f98-9127-248003a7d9a1"
-H "Content-Type: application/json" -d '{
  "name": "My Publish Profile 2.1",
  "webAnalytics": [
    {
      "key" : "sample_key",
      "value" : "sample_value"
    },
    {
      "key" : "sample_key2",
      "value" : "sample_value2"
    }
  ]
}' 'https://api.socialstudio.radian6.com/v3/publishprofiles'
```

After you create a publish macro, add it to a workspace to be used within the application. You can only use publish macros within a single workspace. See [/workspaceresources](#) for more information on how to add a publish macro to a workspace.

Retrieve Publish Macros

This method retrieves publish macros in the Social Studio organization

```
/v3/publishprofiles
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*workspace* - string value indicating the unique ID for the workspace from which to retrieve
publish macros (required)
*page* - integer value indicating which page to display from retrieved data (defaults to
1)
*limit* - integer value indicating the number of results to display per page (defaults to
20)
*orderBy* - string value indicating parameter used to sort results (defaults to date)
*order* - string value indicating display order for retrieved content:
    *desc* - descending order
    *asc* - ascending order (default)
*search* - string value indicating characters to search for in name, returns paginated
results from name matches
```

Request Parameters Example

```
GET /v3/publishProfiles
```

Sample Call

```
curl
'{HOST}/v3/publishProfiles?workspace=028e2d63-0914-11e3-8d6f-168170973bd5&page=2&limit=2'
```

Sample Response

A successful call returns the following response:

```
{
  "publishProfiles": [
    {
      "id": "000ca3cf-3ed1-4b7d-8253-0d50fe1e93fd",
```

```

        "type": "publish",
        "name": "My Publish Profile",
        "description": "testing",
        "owner": 6834,
        "created": 'created_date',
        "socialPropertyIds": '/v3/publishProfiles/:id/socialPropertyIds',
        "youtubePlaylists": '/v3/publishProfiles/:id/youtubePlaylists',
        "targetingIds": '/v3/publishProfiles/:id/targetingIds'
    },
    {
        "id": "000ca3cf-3ed1-4b7d-8253-0d50fe1e93fd",
        "type": "publish",
        "name": "My Publish Profile",
        "description": "testing",
        "owner": 6834,
        "created": 'created_date',
        "socialPropertyIds": '/v3/publishProfiles/:id/socialPropertyIds',
        "targetingIds": '/v3/publishProfiles/:id/targetingIds'
    }
],
"count": 4,
"next": "{HOST}/v3/publishProfiles?limit=2&workspace=:workspaceId&page=2"
}

```

id - string value indicating unique identifier for publish profile
 type - string value indicating type of profile
 publish
 name - string value indicating name of publish profile
 description - string value indicating description of publish profile
 owner - integer value indicating unique ID for publish profile owner
 created - timestamp value indicating date of creation for publish profile
 socialPropertyIds - array of integer values indicating social properties associated with publish profile
 targetingIds - array of string values indicating the social network and the category to target

A bad request returns a 400 error and the following response:

```

{
    "status": 400,
    "code": "error.bad_request",
    "message": "Invalid data supplied."
}

```

A call that finds no publish macros returns the following response:

```

{
    "publishProfiles": []
}

```

Retrieve Publish Macro

This method retrieves information on a publish macro from an account.

```
/v3/publishprofiles/{id}
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - string value indicating unique identifier for publish macro
```

Request Parameters Example

```
GET /v3/publishprofiles/{id}
```

Sample Call

```
GET /v3/publishprofiles/740a25ea-0b57-11e3-8d6f-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  "id": "740a25ea-0b57-11e3-8d6f-168170973bd5",
  "type": "publish",
  "name": "My Publish Profile",
  "description": "testing",
  "owner": 6834,
  "created": 'created_date',
  "socialPropertyIds": '/v3/publishProfiles/:id/socialPropertyIds',
  "targetingIds": '/v3/publishProfiles/:id/targetingIds'
}
```

```
*id* - string value indicating unique identifier for publish macro
*type* - string value indicating type of profile
  *publish*
*name* - string value indicating name of publish macro
*description* - string value indicating description of publish macro
*owner* - integer value indicating unique ID for publish macro owner
*created* - timestamp value indicating date of creation for publish macro
*socialPropertyIds* - array of integer values indicating social properties associated with
  publish macro
```

```
*targetingIds* - array of string values indicating the social network and the category to target
```

Retrieve Publish Macro Attribute

This method retrieves hydrated information on a specified attributes from a publish macro in an account.

```
/v3/publishprofiles/{id}/{type}
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - string value indicating unique identifier for publish macro (required)
*type* - string value indicating property to retrieve:
    *targetingIds* - array of string values indicating the social network and the category to target
    *socialPropertyIds* - array of integer values indicating social properties associated with publish macro
    *webAnalytics* - array of key and value pairs indicating web analytical information to include with publish macro
```

Request Parameters Example

```
GET /v3/publishprofiles/{id}/{type}
```

Sample Call

```
GET /v3/publishprofiles/740a25ea-0b57-11e3-8d6f-168170973bd5/targetingIds
```

Sample Response

A successful call for targeting IDs returns the following response:

```
{
  "targetingIds": [
    "li_geo_af.dz",
    "li_geo_af.eg"
  ],
  "count": 9,
  "next":
    "{HOST}/v3/publishProfiles/0151bc2e-ee3d-4108-b7cf-b3bb0d207b91/targetingIds&page=2"
}
```

A successful call for targeting IDs returns the following response:

```
{
  "webAnalytics": [
    {
      "key": "another one",
      "value": "another value"
    },
    {
      "key": "third one",
      "value": "third value"
    },
    {
      "key": "ytm_dd",
      "value": "ytm_value"
    }
  ],
  "count": 3,
  "next": null
}
```

A bad request returns a 400 error and the following response:

```
{
  "status": 400,
  "code": "error.bad_request",
  "message": "Invalid data supplied."
}
```

A call that finds no publish macro returns the following response:

```
{
  "status": "404",
  "code": "error.not_found",
  "message": "publish macro not found"
}
```

Create Publish Macro

This method creates a publish macro within an account.

```
/v3/publishprofiles
```

HTTP Method

```
POST
```

Request Parameters

```
None
```


JSON Parameters

```
*name* - string value indicating name of publish macro (required)
*description* - string value including description of publish macro (required)
*socialPropertyIds* - array of integer values indicating social properties associated with
publish macro
*youtubePlaylists* - array of key and value pairs indicating the account and playlists
associated with the publish macro
*targetingIds* - array of string values indicating the social network and the category to
target
*webAnalytics* - array of key and value pairs indicating web analytical information to
include with publish macro
```

Request Parameter Example

```
POST /v3/publishprofiles
```

Sample Request

```
curl -X POST {HOST}/v3/publishprofiles
{
  "name": "My Publish Profile",
  "description": "My Publish Profile Description",
  "socialPropertyIds": ['3211', '4567'],
  "youtubePlaylists": [
    {
      "key" : "3211",
      "value" : "11"
    },
    {
      "key" : "3211",
      "value" : "22"
    }
  ],
  "targetingIds": ['li_ag', 'fb_nj'],
  "webAnalytics": [
    {
      "key" : "sample_key",
      "value" : "sample_value"
    },
    {
      "key" : "sample_key2",
      "value" : "sample_value2"
    }
  ]
}
```

Sample Response

A successful call returns an ID value for the newly created publish macro:

```
{
  "id": "5239534f-0b58-11e3-8d6f-168170973bd5"
}
```

A bad request returns a 400 error and the following response:

```
{
  "status": 400,
  "code": "error.bad_request",
  "message": "Invalid data supplied."
}
```

Delete Publish Macro

This method deletes a publish macro from an account.

```
/v3/publishprofiles/{id}
```

HTTP Method

```
DELETE
```

Request Parameters

```
*id* - string value indicating unique identifier for publish macro
```

Request Parameters Example

```
DELETE /v3/publishprofiles/{id}
```

Sample Call

```
curl -X DELETE {HOST}/v3/publishprofiles/740a25ea-0b57-11e3-8d6f-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  true
}
```

A call that finds no publish macro returns the following response:

```
{
  "status": "404",
  "code": "error.not_found",
  "message": "publish macro not found"
}
```

A call from a user without permission to remove the publish macro returns the following response:

```
{
  "status": "403",
  "code": "error.FORBIDDEN",
  "message": "Unauthorized to view profile"
}
```

CHAPTER 13 Approval Rules

In this chapter ...

- [Retrieve Approval Rules](#)
- [Retrieve Approval Rule](#)
- [Create Approval Rule](#)
- [Update Approval Rule](#)
- [Delete Approval Rule](#)
- [Retrieve Approval Rule Criteria](#)
- [Update Approval Rule Criteria](#)
- [Retrieve Approvers for a Rule](#)
- [Update Approvers for a Rule](#)

Approval rules gate actions (such as publishing a post) until certain individuals provide approval. Approval rules are made up of three end-points:

1. `/v1/approvalflows`
2. `/v1/approvalprerequisites`
3. `/v1/approvalprocedures`

Sample Workflow

Approval rules define the name and priority of the rule. Prioritizing rules allows for multi-step approvals. For example, your rule could require that you first get approval from the Creative Team (rule 1), then get approval from the Legal OR from the Director of Social (rule 2).

Define the criteria to trigger the rule within approval prerequisites (such as a user or social account in the workspace).

Finally, approval procedures define the approvers of the rule.

Create a Rule

Let's start by creating a rule, including criteria (prerequisites) and approvers (procedures).

Create an approval rule cURL Example

```
curl -X POST -H "access_token: a78cabb9-c00e-4f98-9127-248003a7d9a1"
-H "Content-Type: application/json" -d '{
  "name": "My Approval Rule",
  "trigger" : "Publish",
  "priority" : "21",
  "workspace" : "b8e4c9c4-9e3c-411f-ae80-7c2b3d6e2ab0"
}' 'https://api.socialstudio.radian6.com/v1/approvalflows'
```

Create the criteria that triggers the rule cURL Example

```
curl -X PUT -H "access_token: a78cabb9-c00e-4f98-9127-248003a7d9a1"
-d '{
  "items": [
```

```

    {
      "criteria": "submitter",
      "operator": "equals",
      "argument": "79799125"
    },
    {
      "criteria": "social account",
      "operator": "equals",
      "argument": "79798915"
    },
    {
      "criteria": "social account",
      "operator": "equals",
      "argument": "{{social_property2}}"
    }
  ],
  "query": "(1 AND 2) OR ( 1 AND 3 )"
}'
'https://api.socialstudio.radian6.com/v1/approvalprerequisites/1d9217cf-40f4-4f8d-b465-9c4542209c20'

```

Define the Approvers of the Rule cURL Example

```

curl -X PUT -H "access_token: 63ae7c90-2927-470c-9ce4-e2489d6e992b"
-H "Content-Type: application/json" -d '{
  "items": [
    {
      "criteria": "user",
      "operator": "equals",
      "argument": "79799125",
      "action": "approve"
    },
    {
      "criteria": "user",
      "operator": "equals",
      "argument": "79799126",
      "action": "approve"
    },
    {
      "criteria": "user",
      "operator": "equals",
      "argument": "79799127",
      "action": "approve"
    }
  ],
  "query": "(1 AND 2) OR ( 1 AND 3 )"
}'
'https://api.socialstudio.radian6.com/v1/approvalprocedures?approvalflow=1d9217cf-40f4-4f8d-b465-9c4542209c20'

```

Retrieve Approval Rules

This method retrieves one or more approval rules for a workspace.

```
/v1/approvalflows
```

HTTP method

```
GET
```

Request Parameters

```
*ownerId* - identifies the workspace from which to retrieve the approval rules (required)
*ownerType* - identifies the object from which to retrieve the approval rules (required).
Possible values include the following:
    *workspace*
*include* - identifies optional additional information on the workflow. Possible values
include the following:
    *prerequisite* - displays array value of rules within approval rule
    *procedure* - returns array value of procedures
    *approverCount* - returns integer value of approvers for approval rule
    *criteriaCount* - returns integer value of criteria for approval rule
*page* - indicates optional number of pages to provide for results
*perpage* - indicates optional number of results to provide per page
```

Request Parameter Example

```
GET {HOST}/v1/approvalflows
```

Sample Request

```
GET /v1/approvalflows?ownerId=a272b824-bbf7-4a8c-bb1e-3d9695f3032a&ownerType=workspace
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": [
    {
      "id": "d460106d-794b-11e3-b8e3-168170973bd5",
      "ownerId": "92ed3d44-5376-11e3-b8e3-168170973bd5",
      "ownerType": "workspace",
      "priority": 1,
      "triggerAction": "publish",
      "name": "My Approval Flow"
    }
  ]
}
```

```

    },
    {
      "id": "b64ee373-81f0-11e3-b8e3-168170973bd5",
      "ownerId": "92ed3d44-5376-11e3-b8e3-168170973bd5",
      "ownerType": "workspace",
      "priority": 2,
      "triggerAction": "publish",
      "name": "My Other Approval Flow"
    }
  ],
  "meta": {
    "total": 2,
    "page": 1,
    "perpage": 2,
    "current":
"HOST/v1/approvalflows?workspace=92ed3d44-5376-11e3-b8e3-168170973bd5&page=1",
    "prev": "",
    "next": ""
  }
}

```

A successful call with include parameters returns the following response:

```

{
  "status": true,
  "response": [
    {
      "id": "d460106d-794b-11e3-b8e3-168170973bd5",
      "ownerId": "92ed3d44-5376-11e3-b8e3-168170973bd5",
      "ownerType": "workspace",
      "priority": 99,
      "triggerAction": "publish",
      "name": "Workflow get with include",
      "prerequisite": {
        "items": [
          {
            "criteria": "submitter",
            "operator": "equals",
            "argument": 7664
          },
          {
            "criteria": "submitter",
            "operator": "equals",
            "argument": 7641
          },
          {
            "criteria": "language",
            "operator": "equals",
            "argument": 60
          },
          {
            "criteria": "language",
            "operator": "equals",
            "argument": 61
          }
        ]
      }
    }
  ]
}

```

```

        ],
        "query": "1 and 2 and ( 3 or 4 )"
    },
    "procedure": [],
    "meta": {
        "criteriacount": "4"
        "approvercount": "1"
    }
}
],
"meta": {
    "total": 1,
    "page": 1,
    "perpage": 1,
    "current":
"HOST/v1/approvalflows?include=prerequisite,procedure&workspace=92ed3d44-5376-11e3-b8e3-168170973bd5&page=1",

    "prev": "",
    "next": ""
}
}

```

status - string value indicating success of call
 response - array of information related to retrieved approval rules:
 id - string value indicating ID of approval rule
 ownerId - string value indicating ID of approval rule owner
 ownerType - string value indicating entity type of approval rule owner
 priority - integer value indicating priority for approval rule
 triggerAction - string value indicating action that begins approval rule
 name - string value indicating name of approval rule
 prerequisites - array indicating rules contained within approval rule:
 criteria - string value indicating object within rule
 operator - string value indicating whether criteria EQUALS or NOT EQUALS the supplied argument value
 argument - string value to compare with criteria
 query - string values indicating AND or OR for query with supplied criteria

Error Responses

A call including an invalid or unsupported argument key and value pair returns the following response:

```

{
    "status": false,
    "response": {
        "errors": [
            {
                "code": "error.bad_request",
                "message": "An invalid parameter/value was supplied"
            }
        ]
    },
    "meta": []
}

```


Ensure that your call includes only valid values.

A call not including an expected argument returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An expected argument was not supplied"
      }
    ]
  },
  "meta": []
}
```

Ensure that your call includes all required values.

A call attempting to view a workspace without the proper permissions returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to view workspace"
      }
    ]
  },
  "meta": []
}
```

Retrieve Approval Rule

This method retrieves one or more approval rules for a workspace.

```
/v1/approvalflows/{id}
```

HTTP method

```
GET
```

Request Parameters

```
*id* - identifies the approval rule to retrieve (required)
*include* - identifies optional additional information on the workflow. Possible values
include the following:
  *prerequisite* - displays array value of rules within approval rule
  *procedure* - returns array value of procedures
```

```

    *approverCount* - returns integer value of approvers for approval rule
    *criteriaCount* - returns integer value of criteria for approval rule
    *hydrate=argument* - indicates that the call should return related entity information instead
    of string or integer IDs

```

Request Parameter Example

```
GET {HOST}/v1/approvalflows/{id}
```

Sample Request

```
GET /v1/approvalflows?ownerId=a272b824-bbf7-4a8c-bb1e-3d9695f3032a&include=prerequisite,procedure,approvercount,criteriaCount&hydrate=argument
```

Sample Response

A successful call returns the following response:

```

{
  "status": true,
  "response": {
    "id": "d460106d-794b-11e3-b8e3-168170973bd5",
    "ownerId": "92ed3d44-5376-11e3-b8e3-168170973bd5",
    "ownerType": "workspace",
    "priority": 171,
    "triggerAction": "publish",
    "name": "My Approval Flow"
  }
}

```

A successful call with include parameters returns the following response:

```

{
  "status": true,
  "response": {
    "id": "e1c69b8c-6844-43fb-9eb3-0fe30dd92cc3",
    "ownerId": "7046e184-0bd1-4c88-9255-642d8a212f31",
    "ownerType": "workspace",
    "priority": 7,
    "triggerAction": "publish",
    "name": "testworkflowapi",
    "prerequisite": {
      "items": [
        {
          "criteria": "submitter",
          "operator": "equals",
          "argument": 6721
        },
        {
          "criteria": "social account",
          "operator": "equals",

```

```

        "argument": "1a4ebec4-7b11-4d0a-aad3-4fc859ba8e28"
      },
      {
        "criteria": "social account",
        "operator": "equals",
        "argument": "d5594362-829d-49c6-b317-6675b9a280f8"
      }
    ],
    "query": "( 1 and 2 ) or 3"
  },
  "procedure": {
    "items": [
      {
        "criteria": "user",
        "operator": "equals",
        "argument": 6721,
        "action": "approve"
      },
      {
        "criteria": "user",
        "operator": "equals",
        "argument": 6721,
        "action": "approve"
      }
    ],
    "query": "1 or 2"
  },
  "meta": {
    "criteriacount": "4"
    "approvercount": "1"
  }
},
"meta": []
}

```

A successful call with include parameters and the hydrate argument returns the following response:

```

{
  "status": true,
  "response": {
    "id": "e1c69b8c-6844-43fb-9eb3-0fe30dd92cc3",
    "ownerId": "7046e184-0bd1-4c88-9255-642d8a212f31",
    "ownerType": "workspace",
    "priority": 7,
    "triggerAction": "publish",
    "name": "testworkflowapi",
    "prerequisite": {
      "items": [
        {
          "criteria": "submitter",
          "operator": "equals",
          "argument": {
            "id": 6721,
            "email": "pboddu@example.com",
            "username": "pboddu@example.com",

```

```

        "title": "PraveenBoddu",
        "display_name": "PraveenBoddu",
        "timezone": "GMT",
        "enabled": true,
        "updated": "2014-02-26T20:14:47+0000",
        "organization_id": 10545
    }
},
{
    "criteria": "social account",
    "operator": "equals",
    "argument": null
},
{
    "criteria": "social account",
    "operator": "equals",
    "argument": null
}
],
"query": "( 1 and 2 ) or 3"
},
"procedure": {
    "items": [
        {
            "criteria": "user",
            "operator": "equals",
            "argument": {
                "id": 6721,
                "email": "pboddu@example.com",
                "username": "pboddu@example.com",
                "title": "PraveenBoddu",
                "display_name": "PraveenBoddu",
                "timezone": "GMT",
                "enabled": true,
                "updated": "2014-02-26T20:14:47+0000",
                "organization_id": 10545
            },
            "action": "approve"
        },
        {
            "criteria": "user",
            "operator": "equals",
            "argument": {
                "id": 6721,
                "email": "pboddu@example.com",
                "username": "pboddu@example.com",
                "title": "PraveenBoddu",
                "display_name": "PraveenBoddu",
                "timezone": "GMT",
                "enabled": true,
                "updated": "2014-02-26T20:14:47+0000",
                "organization_id": 10545
            },
            "action": "approve"
        }
    ]
}

```

```

        }
      ],
      "query": "1 or 2"
    }
  },
  "meta": []
}

```

status - string value indicating success of call
 response - array of information related to retrieved approval rules:
 id - string value indicating ID of approval rule
 ownerId - string value indicating ID of approval rule owner
 ownerType - string value indicating entity type of approval rule owner
 priority - integer value indicating priority for approval flow
 triggerAction - string value indicating action that begins approval rule
 name - string value indicating name of approval rule
 prerequisites - array indicating rules contained within approval rule:
 criteria - string value indicating object within rule
 operator - string value indicating whether criteria EQUALS or NOT EQUALS the
 supplied argument value
 argument - string value to compare with criteria
 query - string values indicating AND or OR for query with supplied criteria

The information provided with the hydrate argument varies depending on the included entities.

Error Responses

An unsuccessful call will return one of the following error responses:

A call including an invalid or unsupported argument key and value pair returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An invalid parameter/value was supplied"
      }
    ]
  },
  "meta": []
}

```

Ensure that your call includes only valid values.

A call not including an expected argument returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An expected argument was not supplied"
      }
    ]
  }
}

```

```

    }
  ]
},
"meta": []
}

```

Ensure that your call includes all required values.

A call attempting to view a workspace without the proper permissions returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to view workspace"
      }
    ]
  },
  "meta": []
}

```

Ensure that you include the proper authentication with your call.

Create Approval Rule

This method creates an approval rule for a workspace.

```
/v1/approvalflows
```

HTTP method

```
POST
```

Request Parameters

```
None
```

Query Parameters

```

*workspace* - ID for the workspace (required)
*trigger* - identifies the event that begins the workflow process. This parameter accepts
the following string values:
  *publish*
*name* - identifies the name of the workflow. This parameter accepts any unique string
value for a specific owner that does not include the < or > characters.
*priority* - accepts an optional integer value that sets the priority for the workflow

```

Request Parameter Example

```
POST {HOST}/v1/approvalflows
```

Sample Request

```
POST /v1/approvalflows HTTP/1.1
Host: HOST
Content-Type: application/x-www-form-urlencoded

trigger=publish&workspace=92ed3d44-5376-11e3-b8e3-168170973bd5&priority=9&name=My+Approval+Flow
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "id": "d460106d-794b-11e3-b8e3-168170973bd5"
  },
  "meta": []
}
```

status - string value indicating success of call
 id - string value indicating ID for newly created approval rule

Error Responses

A call including an invalid or unsupported argument key and value pair returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An invalid parameter/value was supplied"
      }
    ]
  },
  "meta": []
}
```

Ensure that your call includes only valid values.

A call attempting to create or update an entity that already uses the same unique identifiers returns the following response:

```
{
  "status": false,
  "response": {
```

```

    "errors": [
      {
        "code": "error.conflict",
        "message": "A duplicate entry was detected when saving"
      }
    ],
    "meta": []
  }

```

Ensure that your new call includes a unique value.

Missing Argument

A call not including an expected argument returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An expected argument was not supplied"
      }
    ]
  },
  "meta": []
}

```

Ensure that your call includes all required values.

Unauthorized to Add Approval Flow

A call attempting to add an approval rule without the proper permissions returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to remove"
      }
    ]
  },
  "meta": []
}

```

Ensure that you include the proper authentication with your call.

A call attempting to view a non-existent workflow returns the following response:

```

{
  "status": false,
  "response": {

```



```

    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to view workspace"
      }
    ],
    "meta": []
  }
}

```

Ensure that you include the proper authentication with your call.

Update Approval Rule

This method updates an existing approval rule.

```
/v1/approvalflows/{id}
```

HTTP Method

```
PATCH
```

Request Parameters

```

*id* - ID for the approval rule (required)
*name* - identifies the name of the workflow. This parameter accepts any unique string
value for a specific owner that does not include the < or > characters.
*priority* - accepts an optional integer value that sets the priority for the workflow

```

Request Parameter Examples

```
PATCH    /v1/approvalflows/{id}
```

Sample Request

This sample request updates an approval rule with a priority of 2:

```
curl -X PATCH -d "priority=2" {HOST}/v1/approvalflows/d460106d-794b-11e3-b8e3-168170973bd5
```

Sample Response

A successful call returns the following response:

```

{
  "status": true,
  "response": true,
}

```

```

    "meta": []
  }

```

status - string value indicating success of call
 response - string value indicating update of approval rule

Error Responses

An unsuccessful call will return one of the following responses:

A call including an invalid or unsupported argument key and value pair returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An invalid parameter/value was supplied"
      }
    ]
  },
  "meta": []
}

```

Ensure that your call includes only valid values.

A call attempting to create or update an entity that already uses the same unique identifiers returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.conflict",
        "message": "A duplicate entry was detected when saving"
      }
    ]
  },
  "meta": []
}

```

Ensure that your new call includes a unique value.

A call attempting to modify an approval rule without the proper permissions returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to modify"
      }
    ]
  }
}

```

```
    },  
    "meta": []  
}
```

Ensure that you include the proper authentication with your call.

A call attempting to view a non-existent workflow returns the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.forbidden",  
        "message": "Unauthorized to view workspace"  
      }  
    ]  
  },  
  "meta": []  
}
```

Ensure that you include the proper authentication with your call.

Delete Approval Rule

This method deletes an existing approval rule.

```
/v1/approvalflows
```

HTTP method

```
DELETE
```

Request Parameters

```
*id* - ID for the approval rule (required)
```

Request Parameter Examples

```
DELETE /v1/approvalflows/{id}
```

Sample Request

This sample request updates an approval rule with a priority of 2:

```
DELETE {HOST}/v1/approvalflows/d460106d-794b-11e3-b8e3-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": true,
  "meta": []
}
```

status - string value indicating success of call
 response - string value indicating deletion of approval rule

Error Responses

An unsuccessful call will return one of the following responses:

A call including an invalid or unsupported argument key and value pair returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An invalid parameter/value was supplied"
      }
    ]
  },
  "meta": []
}
```

Ensure that your call includes only valid values.

A call not including an expected argument returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": "An expected argument was not supplied"
      }
    ]
  },
  "meta": []
}
```

Ensure that your call includes all required values.

A call attempting to remove an approval rule without the proper permissions returns the following response:

```
{
  "status": false,
```

```
{
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": "Unauthorized to remove"
      }
    ],
    "meta": []
  }
}
```

Ensure that you include the proper authentication with your call.

Retrieve Approval Rule Criteria

This method retrieves criteria that will trigger an approval flow.

```
/v1/approvalprerequisites/{workflowId}
```

HTTP method

```
GET
```

Request Parameters

```
*workflowId* - ID for the approval flow (required)
```

Request Parameter Examples

```
GET /v1/approvalprerequisites/{workflowId}
```

Query Parameters

```
*hydrate=argument - indicates that the call should return related entity information instead of string or integer IDs
```

Sample Request

```
GET {HOST}/v1/approvalprerequisites/46e66860-25fe-11e3-b1a1-168170973bd5
```

```
GET {HOST}/v1/approvalprerequisites/46e66860-25fe-11e3-b1a1-168170973bd5&hydrate=argument
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": {
    "items": [
      {
        "criteria": "submitter",
        "operator": "equals",
        "argument": 123123
      },
      {
        "criteria": "social account",
        "operator": "equals",
        "argument": 83967
      },
      {
        "criteria": "social account",
        "operator": "equals",
        "argument": 84975
      }
    ],
    "query": "(1 AND 2) OR ( 1 AND 3 )"
  },
  "meta": []
}
```

A successful call including the hydrate argument returns the following response:

```
{
  "status": true,
  "response": {
    "items": [
      {
        "criteria": "submitter",
        "operator": "equals",
        "argument": {
          "id": 7664,
          "email": "qzhen@example.com",
          "title": "Qzhen_10100",
          "avatar_url": null,
          "timezone": "GMT",
          "enabled": true,
          "updated": "2014-02-04T15:50:42+0000",
          "organization_id": 10100,
          "roles": 1
        }
      },
      {
        "criteria": "social account",
        "operator": "in",
        "argument": [
          {

```

```

        "name": "Exercise_Your_Voting_Right",
        "id": "541a6b23-21ff-4c2c-9ed0-ca81da98d71d",
        "network": "Facebook",
        "asset": "519296421491495",
        "socialaccount": 94093,
        "tokenStatus": 1,
        "metadata": {
            "pageLogo":
"https://fbcdn-profile-a.akamaihd.net/static-ak/rsrc.php/v2/yg/r/LjGkeBM0Ist.png",
            "provider": "FACEBOOK",
            "userName": "/Exercise_Your_Voting_Right/519296421491495"
        }
    },
    {
        "name": "Example Page",
        "id": "3a65a221-c16e-479b-8430-57b46fcb21f2",
        "network": "Facebook",
        "asset": "407001072724791",
        "socialaccount": 94268,
        "tokenStatus": 1,
        "metadata": {
            "pageLogo":
"https://fbcdn-profile-a.akamaihd.net/static-ak/rsrc.php/v2/yg/r/LjGkeBM0Ist.png",
            "provider": "FACEBOOK",
            "userName": "/Chris-Dev-Page/407001072724791"
        }
    }
]
},
{
    "criteria": "social account",
    "operator": "equals",
    "argument": {
        "id": "83967",
        "link":
"https://stg3-api.dev.cal.sfmco/socialcloud/v2/socialProperties/83967",
        "updated": "2015-04-20T14:45:35Z",
        "title": "Ana-192",
        "topic": {
            "id": "83967",
            "link":
"https://stg3-api.dev.cal.sfmco/socialcloud/v2/topics/83967",
            "updated": "2015-02-19T17:06:49Z",
            "title": "Ana-192",
            "createdDate": "2015-02-19T17:06:49Z"
        },
        "creator": {
            "id": "79798023",
            "link":
"https://stg3-api.dev.cal.sfmco/socialcloud/v2/users/79798023",
            "updated": "2015-04-20T14:45:35Z",
            "title": "Hiren Gosalia",
            "email": "hgosalia@salesforce.com",
            "avatarURL": null
        }
    }
}

```

```

    },
    "projects": [],
    "registration": {
      "id": "16243",
      "link":
"https://stg3-api.dev.cal.sfmt.co/socialcloud/v2/socialAccounts/16243",
      "updated": null,
      "title": "Mansi SFDC",
      "lastRefreshedTime": null
    },
    "lastValidatedTime": null,
    "avatar":
"https://media.licdn.com/mpr/mpr/p/7/005/086/041/07b736a.png",
    "usedBySalesforce": "false",
    "visibility": "private",
    "uniqueName": "ana-192",
    "realName": "Mansi SFDC",
    "socialNetwork": {
      "id": "3880369",
      "name": "LinkedIn Page",
      "type": 25
    },
    "status": 0,
    "tokens": {
      "tokenStatusId": 0,
      "tokenStatusLastUpdate": null
    }
  }
}
},
"query": "1 AND ( 2 OR 3 )"
},
"meta": []
}

```

status - string value indicating success of call
 response - array of information related to retrieved rules:
 items - array indicating information contained within rule:
 criteria - string value indicating object within rule
 operator - string value indicating whether criteria EQUALS or NOT EQUALS the
 supplied argument value
 argument - string value to compare with criteria
 query - string values indicating AND or OR for query with supplied criteria

The information provided with the hydrate argument varies depending on the included entities.

A call that cannot find the indicated approval flow returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {

```



```

        "code": "error.not_found",
        "message": ""
      }
    ]
  },
  "meta": []
}

```

A call that references an unauthorized approval flow returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": ""
      }
    ]
  },
  "meta": []
}

```

Update Approval Rule Criteria

This method updates criteria that will trigger an approval flow.

```
/v1/approvalprerequisites/{approvalflow}
```

HTTP method

```
PUT
```

Request Parameters

```
*approvalflow* - ID for the approval flow (required)
```

Request Parameter Example

```
PUT {HOST}/v1/approvalprerequisites/46e66860-25fe-11e3-b1a1-168170973bd5
```

JSON Parameters

Each JSON payload includes two arguments:

```
*items* - array of criteria to add or update to the approval flow
```

The `*items*` array includes individual items with the following values:

`*criteria*` - string value indicating the subject of the rule. Acceptable values include the following:

`*label*` - label associated with object

`*submitter*` - user ID for person performing submit action

`*social account*` - social property ID associated with the object

`*language*` - language associated with the object

`*country*` - country associated with the object

`*operator*` - string value indicating whether the rule subject equals or does not equal the provided value

`*equals*`

`*not equals*`

`*argument*` - string value to match against the `*criteria*` value. `*argument*` can accept empty values for label, country, and language. `*argument*` cannot accept empty values for `*submitter*` or `*social account*`.

`*query*` - string value indicating how approval workflow handles criteria

The `*query*` argument includes integers, logical operators, and appropriate groupings.

`*integer*` - position of criteria in `*items*` array, beginning with 1

`*logical operators*` - AND or OR value indicating grouping of criteria

`*grouping*` - Use parentheses to group together integers and logical operators. You cannot use AND and OR in the same grouping.

Sample Request

```
PUT {HOST}/v1/approvalprerequisites/46e66860-25fe-11e3-b1a1-168170973bd5
{
  "items": [
    {
      "criteria": "submitter",
      "operator": "equals",
      "argument": 123123
    },
    {
      "criteria": "social account",
      "operator": "equals",
      "argument": 68304
    },
    {
      "criteria": "social account",
      "operator": "equals",
      "argument": 68647
    }
  ],
  "query": "(1 AND 2) OR ( 1 AND 3 )"
}
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
  "response": true,
  "meta": []
}
```

status - string value indicating success of call

response - string value indicating update of approval flow

A call referencing a nonexistant approval flow returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not_found",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

A call not authorized to modify the approval flow returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

A call including incorrect arguments returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.bad_request",
        "message": ""
      }
    ]
  },
}
```

```
"meta": []  
}
```

Retrieve Approvers for a Rule

This method retrieves the criteria that will trigger a specific approval rule.

```
/v1/approvalprocedures/{id}
```

HTTP method

```
GET
```

Request Parameters

id - ID for the approval rule (required)

Request Parameter Examples

```
GET /v1/approvalprocedures/{id}
```

Query Parameters

hydrate=argument - indicates that the call should return related entity information instead of string or integer IDs

Sample Request

This sample request retrieves the specified approval rule:

```
curl GET {HOST}/v1/approvalprocedures/aab75fac-0453-11e3-8d6f-168170973bd5
```

Sample Response

A successful call returns the following response:

```
{  
  "status": true,  
  "response": {  
    "items": [  
      {  
        "criteria": "user",  
        "operator": "equals",  
        "argument": 123123,  
      }  
    ]  
  }  
}
```

```

        "action": "approve"
      },
      {
        "criteria": "user",
        "operator": "equals",
        "argument": 8765,
        "action": "approve"
      },
      {
        "criteria": "user",
        "operator": "equals",
        "argument": 425548144218779,
        "action": "approve"
      }
    ],
    "query": "(1 AND 2) OR ( 1 AND 3 )"
  },
  "meta": []
}

```

A successful call with the hydrate argument returns the following response:

```

{
  "status": true,
  "response": {
    "items": [
      {
        "criteria": "user",
        "operator": "equals",
        "argument": {
          "id": 6721,
          "email": "pboddu@example.com",
          "username": "pboddu@example.com",
          "title": "PraveenBoddu",
          "display_name": "PraveenBoddu",
          "avatar_url": null,
          "timezone": "GMT",
          "enabled": true,
          "updated": "2014-02-07T18:59:47+0000",
          "organization_id": 10545
        },
        "action": "approve"
      },
      {
        "criteria": "user",
        "operator": "equals",
        "argument": {
          "id": 6721,
          "email": "pboddu@example.com",
          "username": "pboddu@example.com",
          "title": "PraveenBoddu",
          "display_name": "PraveenBoddu",
          "avatar_url": null,
          "timezone": "GMT",
          "enabled": true,

```

```

        "updated": "2014-02-07T18:59:47+0000",
        "organization_id": 10545
      },
      "action": "approve"
    },
    {
      "criteria": "user",
      "operator": "equals",
      "argument": {
        "id": 6721,
        "email": "pboddu@example.com",
        "username": "pboddu@example.com",
        "title": "PraveenBoddu",
        "display_name": "PraveenBoddu",
        "avatar_url": null,
        "timezone": "GMT",
        "enabled": true,
        "updated": "2014-02-07T18:59:47+0000",
        "organization_id": 10545
      },
      "action": "approve"
    }
  ],
  "query": "1 and ( 2 or 3 )"
},
"meta": []
}

```

```

*status* - string value indicating success of call
*response* - array of information related to retrieved rules:
  *items* - array indicating information contained within rule:
    *criteria* - string value indicating object within rule
    *operator* - string value indicating whether criteria EQUALS or NOT EQUALS the
supplied argument value
    *argument* - string value to compare with criteria
    *action* - string value indicating action performed via rule
    *query* - string values indicating AND or OR for query with supplied criteria

```

The information provided with the hydrate argument varies depending on the included entities.

Error Responses

A call referencing an unknown approval rule returns the following response:

```

{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.not_found",
        "message": ""
      }
    ]
  }
}

```

```
    },
    "meta": []
}
```

Ensure that you reference the correct ID for the approval rule.

A call referencing an approval rule without authorization returns the following response:

```
{
  "status": false,
  "response": {
    "errors": [
      {
        "code": "error.forbidden",
        "message": ""
      }
    ]
  },
  "meta": []
}
```

Ensure you use the correct authentication information.

Update Approvers for a Rule

This method updates the approvers for a specified approval rule.

```
/v1/approvalprocedures/{id}
```

HTTP method

```
PUT
```

Request Parameters

```
*id* - ID for the approval rule (required)
```

Request Parameter Example

```
PUT /v1/approvalprocedures/{id}
```

JSON Parameters

Each JSON payload includes two arguments:

```
*items* - array of criteria to add or update to the approval rule
```

The **items** array includes individual items with the following values:

criteria - string value indicating this call affects a specified user. Use only the ***user*** value here.

argument - string value to match against the ***criteria*** value. ***argument*** can accept empty values for label, country, and language. ***argument*** cannot accept empty values for ***submitter*** or ***social account***.

action - indicates the action for the rule. Acceptable values include the following:

approve

query - string value indicating how approval workflow handles criteria

The ***query*** argument includes integers, logical operators, and appropriate groupings.

integer - position of criteria in ***items*** array, beginning with 1

logical operators - AND or OR value indicating grouping of criteria

grouping - Use parantheses to group together integers and logical operators. You cannot use AND and OR in the same grouping.

Sample Request

This sample request updates an approval rule with a priority of 2:

```
curl -X PUT {HOST}/v1/approvalprocedures/46e66860-25fe-11e3-b1a1-168170973bd5
{
  "items": [
    {
      "criteria": "user",
      "operator": "equals",
      "argument": 123123,
      "action": "approve"
    },
    {
      "criteria": "user",
      "operator": "equals",
      "argument": 8765,
      "action": "approve"
    },
    {
      "criteria": "user",
      "operator": "equals",
      "argument": 425548144218779,
      "action": "approve"
    }
  ],
  "query": "(1 AND 2) OR ( 1 AND 3 )"
}
```

Sample Response

A successful call returns the following response:

```
{
  "status": true,
```



```
    "response": true,  
    "meta": []  
}
```

status - string value indicating success of call
response - string value indicating update of approval rule

Error Responses

A call referencing an unknown approval rule returns the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.not_found",  
        "message": ""  
      }  
    ]  
  },  
  "meta": []  
}
```

Ensure you used the correct ID for the approval rule.

A call referencing an approval rule without authorization returns the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.forbidden",  
        "message": ""  
      }  
    ]  
  },  
  "meta": []  
}
```

Ensure you use the correct authentication information.

A call referencing a bad request returns the following response:

```
{  
  "status": false,  
  "response": {  
    "errors": [  
      {  
        "code": "error.bad_request",  
        "message": ""  
      }  
    ]  
  },  
}
```

```
    "meta": []  
  }
```

Ensure you properly formatted the information included in the call.

CHAPTER 14 Social Properties

In this chapter ...

- [Retrieve Properties](#)
- [Retrieve Property](#)

The Social Properties endpoint allows you to retrieve a list of social accounts registered in your Social Studio organization. The call returns only those social accounts available to you based on user permissions and role.

Retrieve All Accounts

Users can retrieve a list of all shared and owned accounts.

```
GET v3/socialProperties
```

cURL Example

```
curl -X GET -H "access_token: 99203c04-621c-4b43-99e4-b76e121faddc"  
  
'https://api.socialstudio.radian6.com/v3/socialProperties?view=shared&owned'
```

Retrieve a Single Account

You can also retrieve information on a specific account:

```
GET v3/socialProperties/{id}
```

cURL Example

```
curl -X GET -H "access_token: 99203c04-621c-4b43-99e4-b76e121faddc"  
  
'https://api.socialstudio.radian6.com/v3/socialProperties/6386'
```

Retrieve Properties

This method retrieves social properties the user can access. This object contains information about a managed account topic profile and associated external user account.

```
/v3/socialProperties
```

HTTP Method

```
GET
```

Request Parameters

```
*view* - string value including comma-separated list of properties to return (defaults to OWNED; (only those properties created by requesting user))
  *OWNED* - all social properties created by requesting user
  *SHARED* - all social properties shared with and not owned by requesting user
*types* - integer value indicating return of social properties with specified social network types
  *3* - Twitter
  *4* - Facebook
  *25* - LinkedIn
  *27* - Sina Weibo
  *28* - YouTube
  *29* - Instagram
  *30* - Pinterest
*workspaceGroupId* - reserved for internal use
*limit* - integer value indicating number of results to return (defaults to 25)
*offset* - integer value indicating result to start with in returned results
*status* - string value including comma-separated list of integers indicating return of social properties with specified status values
*invalidAccountsFirst* - Boolean value indicating return of *invalid* values first, then *OK* and *new* values (defaults to false)
*orderBy* - string value indicating fields used to sort returned social properties:
  *title* - string value describing post
  *creator* - string value including creator of post
  *visibility* - string value indicating visibility of post*
  *status* - integer value indicating status of post
*q* - string value including search string query (minimum of three characters), indicates return of social properties with partial case-insensitive match on topic title or creator name
*ids* - string value indicating comma-delimited list of valid social property IDs to return. Only returns visible social properties. The call returns a 404 error if no social properties are available. The call normalizes duplicate values and ignores all other query parameters.
```

Request Parameter Sample

```
GET /v3/socialProperties
```

```
GET /v3/socialProperties?types=4
```

```
{
  "data": [
    {
      "id": "121218",
      "displayName": "Jane's test software",
      "pageUrl": "https://facebook.com/pages/Janes-test-softwar/128544450650919",
      "networkTypeId": 4,
      "avatarUrl":
https://fbcdn-profile-pic.sfo.cdn.fbcdn.net/profile-pic/v/t1.0-1/c1.12.22/s360x528/128544450650919\_128544450650919\_128544450650919\_n.jpg?\_nc\_cat=109&\_nc\_sid=29e3&\_nc\_ohc=5D79B\_ga=4486113665328633056&\_nc\_ht=fbcdn-profile-pic.sfo.cdn.fbcdn.net,
      "slug": "pages/Janes-test-software/128544450650919",
      "valid": true
    },
    {
      "id": "111819",
      "displayName": "I Heart CBC NB",
      "pageUrl": "https://facebook.com/pages/I-Love-CBC-NB/376189652448432",
      "networkTypeId": 4,
      "avatarUrl":
https://fbcdn-profile-pic.sfo.cdn.fbcdn.net/profile-pic/v/t1.0-1/p1.102/s360x360/376189652448432\_376189652448432\_376189652448432\_n.jpg?\_nc\_cat=109&\_nc\_sid=29e3&\_nc\_ohc=5D79B\_ga=4486113665328633056&\_nc\_ht=fbcdn-profile-pic.sfo.cdn.fbcdn.net,
      "slug": "pages/I-Love-CBC-NB/376189652448432",
      "valid": true
    }
  ],
  "meta": {
    "totalCount": 2
  }
}
```

135

Retrieve Property

This method retrieves a single social property from those available for the user to access. This object contains information about a managed account topic profile and associated external user account. The socialPropertyId value equals the managed account topic profile ID.

```
/v3/socialProperties/{id}
```

HTTP Method

```
GET
```

Request Parameters

id - integer value indicating the ID for the social property to retrieve

Request Parameter Sample

```
GET /v3/socialProperties/{socialPropertyId}
```

Sample Request

```
GET /v3/socialProperties/121218
```

Sample Response

```
{
  "data": [
    {
      "id": "121218",
      "displayName": "Jane's test software",
      "pageUrl": "https://facebook.com/pages/Janes-test-softwar/128544450650919",
      "networkTypeId": 4,
      "avatarUrl": "https://imageproxy.fbsbx.com/profilepic/v1.0/f1.3422/5855281254594025880_image18954511437624623e53798_ga_44861136f53283308e68",
      "slug": "pages/Janes-test-software/128544450650919",
      "valid": true
    },
  ],
  "meta": {
    "totalCount": 1
  }
}
```

```
}  
}
```

```
*id* - string value indicating unique ID for social property  
*displayName* - string value containing name of social property  
*pageUrl* - string value containing URL of social property on social network  
*networkTypeId* - integer value indicating social network  
*avatarUrl* - string value containing location of avatar image file  
*slug* - string value containing slug for social property  
*valid* - boolean property indicating whether social media account remains active and valid
```

CHAPTER 15 Create Topic Profile and Retrieving Posts with Keyword Groups

In this chapter ...

- [Retrieve Topic Profiles](#)
- [Retrieve Topic Profile](#)
- [Create Topic Profile](#)
- [Update Topic Profile](#)
- [Delete Topic Profile](#)
- [Update Keyword Groups in a Topic Profile](#)
- [Delete Keyword Groups in a Topic Profile](#)
- [Update Source Filters in a Topic Profile](#)
- [Activate a Topic Profile](#)

Create Topic Profile

Use the topic profile to define the social data that interests you based on a set of parameters consisting of keywords, sources, languages, regions and media types. For example, you could look for a series of English keywords in Twitter posts originating in the United Kingdom.

First, define the topic profile using the following route:

```
POST /v3/topics
```

cURL Example

```
curl -X POST -H 'access_token: your_access_token' -H 'Content-Type: application/json' -d '{
  "title" : "test12345",
  "languages" : [ "1" ],
  "mediaTypes" : [ { "id" : "5" }, { "id" : "1" }, { "id" : "10" } ]
}' https://api.socialstudio.radian6.com/v3/topics
```

Create Keyword Group

Next, create a keyword group that contains the keywords to be monitored. Keywords allow you to define specific terms which should match post content and define which content flows to the topic profile. Social Studio organizes keywords into various keyword groups using the following route:

```
POST /v3/keywordGroups
```

cURL Example

```
curl -X POST -H "access_token: your_access_token" -H "Content-Type: application/json" -d '{
  "title" : "API Test",
  "type" : 2,
  "active" : true,
  "contains" : [{"tokenType" : 1, "value" : "Coca Cola", "meta" :
```



```
"2" }]]  
' https://api.socialstudio.radian6.com/v3/keywordGroups
```

Assign Keyword Group to Topic Profile

After creating a topic profile and a keyword group, you can link them both together using the following API endpoint:

```
PUT /v3/topics{id}/keywordGroups
```

cURL Example

```
curl -X PUT -H "access_token: your_access_token" -H "Content-Type: application/json" -d '[  
  {  
    "id" : "405492"  
  } ]'  
https://api.socialstudio.radian6.com/v3/topics/273382/keywordGroups
```

Retrieving Posts

Once you create the previous calls, you can retrieve matching posts:

```
GET /v3/posts?topics={id}
```

cURL Example

```
curl -X GET -H "access_token: your_access_token" -H "Content-Type: application/json"  
https://api.socialstudio.radian6.com/v3/posts?topics=273382
```

Each client can use a limited amount of active topic profiles. Any call that causes the client to exceed the limit returns a 400 error.

Retrieve Topic Profiles

This method retrieves specific details on topic profiles based on user permissions and role. You cannot use this route to retrieve information on keywords and keyword groups.

```
/v3/topics
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Query Parameters

```
*workspaceGroupId* - reserved for internal use
*limit* - integer value indicating number of results to return (defaults to 50000)
*offset* - integer value indicating result to start with (defaults to 0)
*q* - string value including characters to search for in title (minimum of 3 characters)
*includeQuickSearch* - boolean value indicating whether to include Quick Search topic
profiles in response
*orderBy* - string values indicating how call sorts returned list of topic profiles:
    *title*
    *visibility*
    *status*
    *created* (ordered alphabetically by display name of creator)
    *creation* (topic ordered by topic creation date)
    *ASC* (default)
    *DESC*
*status* - integer values indicating status of topic profile to return
    *0* - disabled
    *1* - purchased or active
    *2* - trial
    *3* - expired
    *4* - draft
```

Request Parameters Example

```
GET /v3/topics
```

Sample Call

```
GET /v3/topics?q="Scott"&includeQuickSearch=false
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "4",
      "title": "Scott Test 3",
      "creator": {
        "id": "3",
        "link": "https://stg2-dmz3.dev.ca1.example.co/socialcloud/v2/users/3",
        "updated": null,
        "title": "scott@example.com (clientId = 1)"
      },
      "visibility": "private",
      "status": 1,
      "emv": 0,
      "languages": [],
      "mediaTypes": [],
      "regions": [],
      "keywordGroups": [],
      "billingCode": "",
      "includeSourceFilters": [],
      "excludeSourceFilters": [],
      "includeAllSourceFilters": [],
      "createdDate": "2009-10-09T19:27:58Z"
    },
    ...
  ],
  "meta" : {
    "totalCount" : 52
  }
}
```

```
*id* - integer value indicating unique ID for topic profile
*title* - string value indicating name of topic profile
*creator* - object containing information on creator of topic profile:
  *id* - integer value indicating unique ID for creator
  *link* - string value indicating URL for location of creator
  *updated* - timestamp value indicating last update for creator (can be null)
  *title* - string value indicating name of creator
*visibility* - string value indicating visibility of topic profile
  *public*
  *private*
*status* - integer values indicating status of topic profile to return
  *0* - disabled
  *1* - purchased or active
  *2* - trial
  *3* - expired
  *4* - draft
*emv* - integer value indicating estimated monthly value for topic profile
*languages* - array of language values associated with topic profile
*mediaTypes* - array of media type values associated with topic profile
```

```
*regions* - array of region values associated with topic profile
*keywordGroups* - array of keyword group values associated with topic profile
*billingCode* - string value of billing code associated with topic profile
*includeSourceFilters* - array of source filters associated with topic profile
*excludeSourceFilters* - array of source filters specifically excluded from topic profile
*includeAllSourceFilters* - array of all source filters available for topic profile
```

Retrieve Topic Profile

This method retrieves a specific topic profile based on user permissions and role. You cannot use this route to retrieve information on keywords and keyword groups.

```
/v3/topics/{id}
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - integer value indicating unique ID for topic profile
```

Request Parameters Example

```
GET /v3/topics{id}
```

Sample Call

```
GET /v3/topics/4
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "4",
      "title": "Scott Test 3",
      "creator": {
        "id": "3",
        "link": "https://stg2-dmz3.dev.ca1.example.co/socialcloud/v2/users/3",
        "updated": null,
        "title": "scott@example.com (clientId = 1)"
      },
      "visibility": "private",
```

```

        "status": 1,
        "emv": 0,
        "languages": [],
        "mediaTypes": [],
        "regions": [],
        "keywordGroups": [],
        "billingCode": "",
        "includeSourceFilters": [],
        "excludeSourceFilters": [],
        "includeAllSourceFilters": [],
        "createdDate": "2009-10-09T19:27:58Z"
    },
    ...
],
"meta" : {
    "totalCount" : 52
}
}

```

```

*id* - integer value indicating unique ID for topic profile
*title* - string value indicating name of topic profile
*creator* - object containing information on creator of topic profile:
    *id* - integer value indicating unique ID for creator
    *link* - string value indicating URL for location of creator
    *updated* - timestamp value indicating last update for creator (can be null)
    *title* - string value indicating name of creator
*visibility* - string value indicating visibility of topic profile
    *public*
    *private*
*status* - integer values indicating status of topic profile to return
    *0* - disabled
    *1* - purchased or active
    *2* - trial
    *3* - expired
    *4* - draft
*emv* - integer value indicating estimated monthly value for topic profile
*languages* - array of language values associated with topic profile
*mediaTypes* - array of media type values associated with topic profile
*regions* - array of region values associated with topic profile
*keywordGroups* - array of keyword group values associated with topic profile
*billingCode* - string value of billing code associated with topic profile
*includeSourceFilters* - array of source filters associated with topic profile
*excludeSourceFilters* - array of source filters specifically excluded from topic profile
*includeAllSourceFilters* - array of all source filters available for topic profile

```

Create Topic Profile

This method creates a new keyword topic profile.

```
/v3/topics
```

HTTP Method

POST

Request Parameters

None

Request Parameters Example

POST /v3/topics

JSON Parameters

title - string value indicating name for new topic profile (required)
languages - array of integer values indicating languages associated with new topic profile (required, set with value of ALL to associate with all available values)
mediaTypes - array of integer values indicating media types associated with new topic profile (required, set with value of ALL to associate with all available values)
regions - array of integer values indicating regions associated with new topic profile (required, set with value of ALL to associate with all available values)
status - integer value indicating status of topic profile (create with value of 4 for DRAFT, otherwise leave blank)

Sample Request

```
POST /v3/topics
{
  "title": "test topic profile",
  "languages": [
    "1"
  ],
  "mediaTypes": [
    {
      "id": "5"
    },
    {
      "id": "1"
    },
    {
      "id": "10"
    }
  ]
  "regions": [
    "all"
  ]
}
```

Sample Response

A successful call returns a 201 Success code with a Location response header including the URL for the new topic profile resource.

A call including a malformed JSON payload or invalid IDs (such as unauthorized mediatypes) returns a 400 code. Ensure your JSON payload contains valid values.

A call by a user with insufficient permissions returns a 403 code. Ensure your user can make this call within your organization.

An internal server error returns a 500 code. Try your code at a later time.

Update Topic Profile

This method updates a new keyword topic profile.

```
/v3/topics
```

HTTP Method

```
PUT
```

Request Parameters

```
None
```

Request Parameters Example

```
PUT /v3/topics
```

JSON Parameters

```
*title* - string value indicating name for new topic profile (required)
*languages* - array of integer values indicating languages associated with new topic profile
             (required, set with value of ALL to associate with all available values)
*mediaTypes* - array of integer values indicating media types associated with new topic
              profile (required, set with value of ALL to associate with all available values)
*regions* - array of integer values indicating regions associated with new topic profile
            (required, set with value of ALL to associate with all available values)
*status* - integer value indicating status of topic profile (create with value of 4 for
            DRAFT, otherwise leave blank)
```

Sample Request

```
curl -X PUT -H "access_token: f0f59c64-a82d-4a76-8781-078952df8785" -H "Content-Type:
application/json" -d '{
  "title": "API Test TP",
  "languages": [
```

```
    "1"
  ],
  "mediaTypes": [
    {
      "id": "5"
    },
    {
      "id": "1"
    },
    {
      "id": "10"
    },
    {
      "id": "12"
    }
  ]
} 'https://{host}/v3/topics/1228'
```

Sample Response

A successful call returns a 200 Success code.

A call including a malformed JSON payload or invalid IDs (such as unauthorized mediatypes) returns a 400 code. Ensure your JSON payload contains valid values.

A call by a user with insufficient permissions returns a 403 code. Ensure your user can make this call within your organization.

An internal server error returns a 500 code. Try your code at a later time.

Delete Topic Profile

This method deletes an existing topic profile.

```
/v3/topics/{id}
```

HTTP Method

```
DELETE
```

Request Parameters

id - integer value indicating unique ID for topic profile

Request Parameters Example

```
DELETE /v3/topics/{id}
```


Sample Request

```
DELETE /v3/topics/123459
```

Sample Response

A successful call returns a 204 No Content code.

A call including an invalid ID value returns a 404 code. Ensure your call contains valid values.

An internal server error returns a 500 code. Try your code at a later time.

Update Keyword Groups in a Topic Profile

This method associated an existing keyword group with the specified topic profile.

```
/v3/topics/{topicProfileId}/keywordGroups
```

HTTP Method

```
PUT
```

Request Parameters

topicProfileId - integer value indicating the topic profile

Request Parameters Example

```
PUT /v3/topics/{id}/keywordGroups
```

JSON Parameters

keywordGroupId - integer value indicating the keyword group (can include array of these values)

Sample Request

```
curl -X PUT -H "Content-Type: application/json" -H "Accept: application/json"-H "access_token: your_access_token" -d "[
  { \"id\" : \"408299\" }
]" "https://{host}/v3/topics/304237/keywordGroups"
```

Sample Response

A successful call returns a 200 code.

A call from a user with insufficient permissions to perform the update (either on topic profile or keyword groups) returns a 403 call. Contact your administrator for the proper permissions.

A call including an invalid topic profile ID or keyword group ID returns a 404 error. Ensure you include the correct values in your call.

An internal server error during your call returns a 500 error. Try your call again at a later time.

Delete Keyword Groups in a Topic Profile

This method deletes a keyword group existing topic profile.

```
/v3/topics/{topicId}/keywordGroups/{keywordGroupId}
```

HTTP Method

```
DELETE
```

Request Parameters

```
*topicId* - integer value indicating unique ID for topic profile  
*keywordGroupId* - integer value indicating unique UD for keyword group
```

Request Parameters Example

```
DELETE /v3/topics/{id}/keywordGroups/{keywordGroupId}
```

Sample Request

```
DELETE /v3/topics/123459/keywordGroups/54321
```

Sample Response

A successful call returns a 204 No Content code.

A call including an invalid ID value returns a 404 code. Ensure your call contains valid values.

An internal server error returns a 500 code. Try your code at a later time.

Update Source Filters in a Topic Profile

This method updates the source filters associated with a topic profile.

```
/v3/topics/{id}/sourceFilters
```

HTTP method

PUT

Request Parameters

id - integer value indicating unique ID for topic profile

Request Parameters Example

```
PUT /v3/topics/{id}/sourceFilters
[
  {
    "sourceFilterId":2,
    "type":1
  }
]
```

JSON Parameters

sourceFilterId - The identifier for the source filter to be associated with the topic profile
type - Source filter type
 0 - Include
 1 - Exclude
 2 - Include all

Sample Requests

Sample Response

Activate a Topic Profile

This method activates an existing topic profile.

/v3/topics/{id}

HTTP Method

POST

Request Parameters

```
*id* - integer value indicating unique ID for topic profile
```

Request Parameters Example

```
POST /v3/topics/{id}
```

Query Parameters

```
*action* - string value indicating action to perform on topic profile (required)
    *activate*
*billingCode* - string value indicating billing code to associated with activation (required)
```

Sample Request

```
POST /v3/topics/12345?action=activate&billingCode=6789
```

Sample Response

A successful call returns a 204 No Content code.

A call including an invalid ID value returns a 404 code. Ensure your call contains valid values.

An internal server error returns a 500 code. Try your code at a later time.

CHAPTER 16 Keyword Groups

In this chapter ...

- Retrieve Keyword Groups
- Create Keyword Groups
- Update Keyword Groups

Retrieve Keyword Groups

This method returns a specified keyword group from a workspace.

```
/v3/keywordGroups/{id}
```

HTTP Method

```
GET
```

Request Parameters

id - integer value indicting unique ID for keyword group

Request Parameters Example

```
GET /v3/keywordGroups/{id}
```

Sample Request

```
curl -X GET -H "Accept: application/json"-H "access_token: your_access_token"  
"https://{host}/v3/keywordGroups/408299"
```

Sample Response

A successful call returns the following response:

```
{  
  "data" : [  
    {  
      "title" : "Beverages",  
      "type" : 2,  
      "active" : true,  
      "contains" : [  
        [  
          {  
            "tokenType": 1,  
            "value": "soda",  
            "meta": "2"  
          },  
          {  
            "tokenType": 2  
          },  
          {  
            "tokenType": 0,  
            "value": "orange"  
          }  
        ]  
      ]  
    }  
  ]  
}
```

```

],
[
  {
    "tokenType": 1,
    "value": "coffee",
    "meta": "3"
  },
  {
    "tokenType": 3
  },
  {
    "tokenType": 0,
    "value": "tea"
  }
]
],
"excludes": [
  {
    "tokenType": 0,
    "value": "lemonade"
  }
],
"andContains": [
  {
    "tokenType": 0,
    "value": "cranberry"
  },
  {
    "tokenType": 0,
    "value": "water"
  }
]
}
],
"meta" : {
  "totalCount" : 1
}
}

```

title - string value indicating the name of the keyword group

type - integer value indicating the type of the keyword group

active - boolean value indicating whether the keyword group remains active in the account

contains - array of filters that include complex filter queries with operators AND and NOT:

tokenType - integer value indicating types of keyword tokens:

0 - KEYWORD_OR_PHRASE (requires tokenType and value information)

1 - PROXIMITY (requires tokenType, value, and meta information)

2 - AND (requires only tokenType value)

3 - NOT (requires only tokenType value)

value - string value indicating keyword or phrase to include in search

meta - integer value in range of -1 to 20 indicating proximity of key terms to trigger inclusion

andContains - array of values indicating other keywords to include in results (no operators

```

allowed)
  *tokenType* - integer value indicating types of keyword tokens:
    *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
  *value* - string value indicating keyword or phrase to include in search
*excludes* - array of values indicating other keywords to exclude from results (no operators
allowed, cannot exceed the maximum number set at the client level)
  *tokenType* - integer value indicating types of keyword tokens:
    *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
  *value* - string value indicating keyword or phrase to include in search

```

Create Keyword Groups

This method creates a new keyword group not associated with a topic profile.

```
/v3/keywordGroups
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Request Parameters Example

```
GET /v3/keywordGroups
```

JSON Parameters

Each JSON payload includes the following arguments:

```

*title* - string value indicating the name of the keyword group
*type* - integer value indicating the type of the keyword group
*active* - boolean value indicating whether the keyword group remains active in the account
*contains* - array of filters that include contplex filter queries with operators AND and
NOT:
  *tokenType* - integer value indicating types of keyword tokens:
    *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
    *1* - PROXIMITY (requires tokenType, value, and meta information)
    *2* - AND (requires only tokenType value)
    *3* - NOT (requires only tokenType value)
  *value* - string value indicating keyword or phrase to include in search
  *meta* - integer value in range of -1 to 20 indicating proximity of key terms to trigger
inclusion
*andContains* - array of values indicating other keywords to include in results (no operators
allowed)

```



```

*tokenType* - integer value indicating types of keyword tokens:
    *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
*value* - string value indicating keyword or phrase to include in search
*excludes* - array of values indicating other keywords to exclude from results (no operators
allowed, cannot exceed the maximum number set at the client level)
*tokenType* - integer value indicating types of keyword tokens:
    *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
*value* - string value indicating keyword or phrase to include in search

```

Your keywords cannot accept predefined stop words, including the following values:

```

abcdefghijklmnopqrstuvwxyz1234567890!@#$%^&*()a man a plan a canal panama if it ain't broke don't fix it the quick brown fox jumps over the lazy dog

```

Your keywords cannot accept any invalid characters, including the following values:

```

^"<>|*\n

```

The overall number of keywords and phrases for the topic profile cannot exceed the maximum number set at the client level. Contact your administrator to determine these levels.

Sample Request

```

curl -X POST -H "Content-Type: application/json" -H "Accept: application/json"-H
"access_token: your_access_token" -d "{
  \"title\" : \"Presidential Test\",
  \"type\" : 2,
  \"active\" : true,
  \"contains\" : [
    [
      {
        \"tokenType\" : 1,
        \"value\" : \"President\",
        \"meta\" : \"2\"
      }
    ]
  ]
}" "https://{host}/v3/keywordGroups"

```

Sample Response

A successful call returns the following response:

```

{
  "data": [
    {
      "id": "408299"
    }
  ],
  "meta":
  {
    "totalCount": 1
  }
}

```

```
}
}
```

data - array of returned information from the call
id - integer value indicating the unique ID for the created keyword group

A call with a malformed JSON document returns a 400 error. Ensure you correctly formatted your JSON information.

An internal server error during your call returns a 500 error. Try your call again at a later time.

Update Keyword Groups

This method creates a new keyword group not associated with a topic profile.

```
/v3/keywordGroups/{id}
```

HTTP Method

```
PUT
```

Request Parameters

id - integer value indicating unique ID for keyword group

Request Parameters Example

```
PUT /v3/keywordGroups
```

JSON Parameters

Each JSON payload includes the following arguments:

```
*title* - string value indicating the name of the keyword group
*type* - integer value indicating the type of the keyword group
*active* - boolean value indicating whether the keyword group remains active in the account
*contains* - array of filters that include complex filter queries with operators AND and NOT:
    *tokenType* - integer value indicating types of keyword tokens:
        *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
        *1* - PROXIMITY (requires tokenType, value, and meta information)
        *2* - AND (requires only tokenType value)
        *3* - NOT (requires only tokenType value)
    *value* - string value indicating keyword or phrase to include in search
    *meta* - integer value in range of -1 to 20 indicating proximity of key terms to trigger inclusion
*andContains* - array of values indicating other keywords to include in results (no operators
```

```

allowed)
    *tokenType* - integer value indicating types of keyword tokens:
        *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
    *value* - string value indicating keyword or phrase to include in search
*excludes* - array of values indicating other keywords to exclude from results (no operators
allowed, cannot exceed the maximum number set at the client level)
    *tokenType* - integer value indicating types of keyword tokens:
        *0* - KEYWORD_OR_PHRASE (requires tokenType and value information)
    *value* - string value indicating keyword or phrase to include in search

```

Your keywords cannot accept predefined stop words, including the following values:

`abcdefghijklmnopqrstuvwxyz1234567890!@#$%^&*()an asat heby obgh eif,in is it ren ym d,m n,r so t up us we ad th e fo r m,t sh ew as test`

Your keywords cannot accept any invalid characters, including the following values:

$$\wedge \text{ " } \langle \rangle \mid * \backslash n$$

The overall number of keywords and phrases for the topic profile cannot exceed the maximum number set at the client level. Contact your administrator to determine these levels.

Sample Request

```
curl -X PUT -H "Content-Type: application/json" -H "Accept: application/json"-H
"access_token: your_access_token" -d "{
  \"title\": \"Presidential Test\",
  \"active\": true,
  \"type\": 2,
  \"keywordsCount\": 0,
  \"contains\": [
    [
      {
        \"tokenType\": 1,
        \"value\": \"President\",
        \"meta\": \"2\"
      }
    ]
  ],
  \"andContains\": [],
  \"excludes\": []
}" "https://{host}/v3/keywordGroups/408299"
```

Sample Response

A successful call returns a 200 code.

A call with a malformed JSON document returns a 400 error. Ensure you correctly formatted your JSON information.

A call from a user with insufficient permissions to perform the update returns a 403 call. Contact your administrator for the proper permissions.

An internal server error during your call returns a 500 error. Try your call again at a later time.

CHAPTER 17 Sources and Source Groups

In this chapter ...

- [Retrieve Sources](#)
- [Retrieve Source](#)
- [Create Source](#)
- [Update Source](#)
- [Delete Source](#)
- [Retrieve Source Groups](#)
- [Retrieve Source Group](#)
- [Create Source Group](#)
- [Update Source Group](#)
- [Delete Source Group](#)

Use sources to identify specific Facebook pages, Twitter accounts, blog posts, and other source types. These source types can further refine the topic profile definition. You can also combine sources into groups.

Retrieve Sources

```
GET /v3/sourceFilters/1055/sourceFilterQueries
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e"
-H "Content-Type:
application/json" "https://api.socialstudio.radian6.com/v3/sourceFilters/1055/sourceFilterQueries"
```

Retrieve Sources

This method returns all source filter queries associated with a social filter.

```
/v3/sourceFilters/{sourceFilterId}/sourceFilterQueries
```

HTTP Method

```
GET
```

Request Parameters

sourceFilterId - integer value indicating the source filter from which to retrieve a social filter query (required)

Query Parameters

limit - integer value indicating number of results to return (defaults to 5000)
offset - integer value indicating result to start with (defaults to 0)
q - string value including characters to search for in title (minimum of 3 characters)
orderBy - string values indicating how call sorts returned list of source filters:
 id - sort by ID value
 title - sort by title
 uri - sort by URI

Request Parameter Example

```
GET /v3/sourceFilters{id}/sourceFilterQueries
```

Sample Request

```
GET /v3/sourceFilters/1/sourceFilterQueries
```

Sample Response

```
{
  "data": [
    {
      "id": "1",
      "title": "Source Filter Query Title",
      "uri": "my.source.filter.query.org"
    }
  ],
  "meta": {
```

```
    "totalCount": 1
  }
}
```

id - integer value indicating unique identifier for source filter
title - string value indicating name for source filter
uri - string value including source filter query location in system

Retrieve Source

This method returns a specified source filter query associated with a social filter.

```
/v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

HTTP Method

```
GET
```

Request Parameters

sourceFilterId - integer value indicating the source filter from which to retrieve a social filter query (required)
sourceFilterQueryId - integer value indicating the source filter query to retrieve (required)

Query Parameters

limit - integer value indicating number of results to return (defaults to 5000)
offset - integer value indicating result to start with (defaults to 0)
q - string value including characters to search for in title (minimum of 3 characters)
orderBy - string values indicating how call sorts returned list of source filters:
 id - sort by ID value
 title - sort by title
 uri - sort by URI

Request Parameter Example

```
GET /v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

Sample Request

```
GET /v3/sourceFilters/1/sourceFilterQueries/123
```

Sample Response

```
{
  "data": [
    {
      "id": "123",
      "title": "Source Filter Query Title",
      "uri": "my.source.filter.query.org"
    }
  ],
  "meta": {
    "totalCount": 1
  }
}
```

id - integer value indicating unique identifier for source filter
title - string value indicating name for source filter
uri - string value including source filter query location in system

Create Source

This method creates a source filter query within an account. You can add this source filter query to the source filter to create conditions for use within the filter.

```
/v3/sourceFilters/{sourceFilterId}/sourceFilterQueries
```

HTTP Method

```
POST
```

Request Parameters

sourceFilterId - integer value indicating the source filter from which to retrieve a social filter query (required)

Request Parameter Example

```
POST /v3/sourceFilters/{sourceFilterId}/sourceFilterQueries
```

JSON Parameters

title - string value indicating name for source filter
uri - string value indicating location for source filter query(required)

Sample Request

```
POST    /v3/sourceFilters/1/sourceFilterQueries
{
  "title": "Source Filter Title",
  "uri": "A description of what this source filter contains."
}
```

Sample Response

A successful call returns a 201 Success response. A Location response header contains the URL to the new source filter query resource.

A malformed JSON payload returns a 400 Bad Request response. Ensure your payload matches the sample shown in this example.

An internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Update Source

This method updates a source filter query within an account.

```
/v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

HTTP Method

```
PUT
```

Request Parameters

```
*sourceFilterId* - integer value indicating the source filter from which to retrieve a
social filter query (required)
*sourceFilterQueryId* - integer value indicating the source filter query to retrieve
(required)
```

Request Parameter Example

```
PUT /v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

JSON Parameters

```
*title* - string value indicating name for source filter
*uri* - string value indicating location for source filter query(required)
```


Sample Request

```
PUT /v3/sourceFilters/1
{
  "title": "Source Filter Title",
  "uri": "A description of what this source filter contains."
}
```

Sample Response

A successful call returns a 204 No Content response, indicating the call successfully updated the source filter.

A malformed JSON payload returns a 400 Bad Request response. Ensure your payload matches the sample shown in this example.

An internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Delete Source

This method deletes the source filter query specified by the ID. This action removes the source query filter from any associated source filter.

```
/v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

HTTP Method

```
DELETE
```

Request Parameters

```
*sourceFilterId* - integer value indicating the source filter from which to retrieve a
social filter query (required)
*sourceFilterQueryId* - integer value indicating the source filter query to retrieve
(required)
```

Request Parameter Example

```
DELETE /v3/sourceFilters/{sourceFilterId}/sourceFilterQueries/{sourceFilterQueryId}
```

Sample Request

```
DELETE /v3/sourceFilters/1/socialFilterQueries/123
```

Sample Response

A successful call returns a No Content 204 code, indicating the call succeeded and no content exists.

An internal error occurring during the call returns a 500 error.

Retrieve Source Groups

This method returns all source groups available to the authenticated users.

```
/v3/sourceFilters
```

HTTP Method

```
GET
```

Request Parameters

```
*limit* - integer value indicating number of results to return (defaults to 5000)
*offset* - integer value indicating result to start with (defaults to 0)
*q* - string value including characters to search for in title (minimum of 3 characters)
*title* - string value including characters to search for an exact title match
*status* - string value including comma-separated list of status values to search:
    *0* - deactivated source groups
    *2* - active source groups
    *3* - draft source groups
*orderBy* - string values indicating how call sorts returned list of source groups:
    *id* - sort by ID value
    *title* - sort by title
    *visibility* - sort by visibility
    *status* - sort by status
    *creation* - sort by creation date
    *description* - sort by description
    *creator* - sort by creator display name
    *ASC* - sort in ascending order
    *DESC* - sort in descending order
```

Request Parameter Example

```
GET /v3/sourceFilters
```

Sample Request

```
GET /v3/sourceFilters?status=2
```

Sample Response

```
{
  "data": [
    {
```

```

    "id": "1",
    "title": "Trent - Yahoo Answers SF",
    "description": null,
    "status": 2,
    "createdDate": "2009-12-22T17:22:08Z",
    "topicCount": 0,
    "public": false,
    "creator": {
      "id": "58",
      "link": "https://api.radian6.com/socialcloud/v2/users/58",
      "updated": null,
      "title": "Trent - Staging - GMail - CLID = 1"
    }
  },
  ...
],
  "meta": {
    "totalCount": 706
  }
}

```

id - integer value indicating unique identifier for source group
 title - string value indicating name for source group
 description - string value including text describing source group
 status - integer value indicating status of source group
 createdDate - timestamp value indicating date of creation for source group
 topicCount - integer value indicating topics associated with source group
 public - boolean value indicating whether the source group can be viewed publicly
 creator - object including information on user who created the source group
 id - integer value indicating unique identifier for user
 link - string value including URL for user
 updated - timestamp value indicating last time user updated their information (can be null)
 title - string value indicating name of user

Retrieve Source Group

This method returns a source group specified by the ID value.

```
/v3/sourceFilters/{sourceFilterId}
```

HTTP Method

```
GET
```

Request Parameters

sourceFilterId - integer value indicating ID for source group

Request Parameter Example

```
GET /v3/sourceFilters/{sourceFilterId}
```

Sample Request

```
GET /v3/sourceFilters/1
```

Sample Response

```
{
  "data": [
    {
      "id": "1",
      "title": "Trent - Yahoo Answers SF",
      "description": null,
      "status": 2,
      "createdDate": "2009-12-22T17:22:08Z",
      "topicCount": 0,
      "public": false,
      "creator": {
        "id": "58",
        "link": "https://api.radian6.com/socialcloud/v2/users/58",
        "updated": null,
        "title": "Trent - Staging - GMail - CLID = 1"
      }
    },
    ,
    ,
    "meta": {
      "totalCount": 1
    }
  ]
}
```

```
*id* - integer value indicating unique identifier for source group
*title* - string value indicating name for source group
*description* - string value including text describing source group
*status* - integer value indicating status of source group
*createdDate* - timestamp value indicating date of creation for source group
*topicCount* - integer value indicating topics associated with source group
*public* - boolean value indicating whether the source group can be viewed publicly
*creator* - object including information on user who created the source group
  *id* - integer value indicating unique identifier for user
  *link* - string value including URL for user
  *updated* - timestamp value indicating last time user updated their information (can
be null)
  *title* - string value indicating name of user
```

Create Source Group

This method creates a source group within an account. You can then add sources to the source group and create conditions for use within the topic profile.

```
/v3/sourceFilters
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Request Parameter Example

```
POST /v3/sourceFilters
```

JSON Parameters

```
*title* - string value indicating name for source group (create)
*description* - string value including text describing source group (create)
*public* - boolean value indicating whether the source group includes a public status
```

Sample Request

```
POST    /v3/sourceFilters/
{
  "title": "source group title",
  "description": "A description of what this source group contains.",
  "public": false
}
```

Sample Response

A successful call returns a 201 Success response. A Location response header contains the URL to the new source group resource.

A malformed JSON payload returns a 400 Bad Request response. Ensure your payload matches the sample shown in this example.

An internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Update Source Group

This method updates a source group available to an authenticated user.

```
/v3/sourceFilters/{sourceFilterId}
```

HTTP Method

```
PUT
```

Request Parameters

sourceFilterID - The ID for the specific source group you wish to update

Request Parameter Example

```
PUT /v3/sourceFilters/{sourceFilterID}
```

JSON Parameters

title - string value indicating name for source group (required)
description - string value including text describing source group (required)
public - boolean value indicating whether the source group includes a public status

Sample Request

```
PUT /v3/sourceFilters/1
{
  "title": "source group Title",
  "description": "A description of what this source group contains.",
  "public": false
}
```

Sample Response

A successful call returns a 204 No Content response, indicating the call successfully updated the source group.

A malformed JSON payload returns a 400 Bad Request response. Ensure your payload matches the sample shown in this example.

An internal error during creation returns a 500 Internal Server Error. Try your create call again at a later time.

Delete Source Group

This method deletes the source group specified by the ID.

```
/v3/sourceFilters/{sourceFilterId}
```

HTTP Method

```
DELETE
```

Request Parameters

sourceFilterId - integer value indicating ID for source group

Request Parameter Example

```
DELETE /v3/sourceFilters/{sourceFilterId}
```

Sample Request

```
DELETE /v3/sourceFilters/1
```

Sample Response

A successful call returns a No Content 204 code, indicating the call succeeded and no content exists.

An internal error occurring during the call returns a 500 error.

CHAPTER 18 Languages

In this chapter ...

- [Retrieve Languages](#)
- [Retrieve Language](#)

Retrieve Languages

This method retrieves information about all languages available when managing your topic profile.

```
/v3/languages
```

HTTP Method

```
GET
```

Request Parameters

limit - integer value indicating number of results to return (defaults to 1000). This method does not return any deleted comments.
offset - integer value indicating result to start with (defaults to 0)

Request Parameter Example

```
GET /v3/languages
```

Sample Request

```
GET /v3/languages
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e"  
'https://api.socialstudio.radian6.com/v3/languages'
```

Sample Response

A successful call returns the following response:

```
{  
  "data": [  
    {  
      "id": "1",  
      "title": "English",  
      "code": "en"  
    },  
    {  
      "id": "2",  
      "title": "French",  
      "code": "fr"  
    },  
    {  
      "id": "3",  
      "title": "German",  
      "code": "de"  
    }  
  ]  
}
```

```
    "id": "3",
    "title": "Spanish",
    "code": "es"
  },
  ...
],
"meta": {
  "totalCount": 28
}
```

id - integer value identifying the language

title - string value containing the English-language name for the language

code - string value containing the ISO language code for the language

Retrieve Language

This method retrieves information about all languages available within an account.

```
/v3/languages/{id}
```

HTTP Method

```
GET
```

Request Parameters

id - integer value indicating the language to retrieve

Request Parameter Example

```
GET /v3/languages/{id}
```

Sample Request

```
GET /v3/languages/1
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "1",
```

```
    "title": "English",  
    "code": "en"  
  },  
],  
"meta": {  
  "totalCount": 1  
}  
}
```

id - integer value identifying the language
title - string value containing the English-language name for the language
code - string value containing the ISO language code for the language

CHAPTER 19 Regions

In this chapter ...

- [Retrieve Regions](#)
- [Retrieve Region](#)

Retrieve Regions

This method retrieves information about all regions available when managing your topic profile.

/v3/regions

HTTP Method

GET

Request Parameters

```
*limit* - integer value indicating number of results to return (defaults to 1000)
*offset* - integer value indicating result to start with (defaults to 0)
```

Request Parameter Example

```
GET /v3/regions
```

Sample Call

```
GET /v3/regions
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e" -H "Content-Type: application/json" 'https://api.socialstudio.radian6.com/v3/regions'
```

Sample Response

```
{
  "data": [
    {
      "id": "46",
      "title": "China",
      "regionTLD": ".cn",
      "isoCountryCode": "CN"
    },
    {
      "id": "78",
      "title": "French Southern Territories",
      "regionTLD": ".tf",
      "isoCountryCode": "TF"
    },
    {

```

```
    "id": "158",
    "title": "New Caledonia",
    "regionTLD": ".nc",
    "isoCountryCode": "NC"
  },
  ...
],
"meta": {
  "totalCount": 249
}
```

id - integer value indicating internal ID for region
title - string value including English name for region
regionTLD - string value including top-level domain for region
isoCountryCode - string value including ISO country code

Retrieve Region

This method returns a specific region associated with an account.

```
/v3/regions/{regionId}
```

HTTP Method

```
GET
```

Request Parameters

regionId - integer value indicating internal ID for region

Request Parameter Example

```
GET /v3/regions/{regionId}
```

Sample Call

```
GET /v3/regions/46
```

Sample Response

```
{
  "data": [
    {
```

```
    "id": "46",
    "title": "China",
    "regionTLD": ".cn",
    "isoCountryCode": "CN"
  }
],
"meta": {
  "totalCount": 1
}
}
```

id - integer value indicating internal ID for region
title - string value including English name for region
regionTLD - string value including top-level domain for region
isoCountryCode - string value including ISO country code

CHAPTER 20 Media Types

In this chapter ...

- [Retrieve Media Types](#)
- [Retrieve Media Type](#)

Retrieve Media Types

This method retrieves a collection of media types on supplied parameters.

```
/v3/mediaTypes
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Request Parameter Example

```
GET /v3/mediaTypes
```

Query Parameters

```
*private* - boolean value indicating a return of only private media types (defaults to false)
*classificationIds* - string value containing comma-delimited list of media type classification IDs. This parameter returns media types belonging to the specified classifications. When supplying this value, call will ignore *private* parameter and return all results associated with supplied classification IDs.
*limit* - integer value indicating number of results to return (defaults to 1000)
*offset* - integer value indicating result to start with in returned results
```

Sample Request

```
GET /v3/mediaTypes?private=true&classificationIds=1,3
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fdle"
'https://api.socialstudio.radian6.com/v3/mediaTypes'
```

Sample Response

```
{
  "data": [
    {
      "id": "1",
```

```
{
  "title": "Blogs",
  "isVisible": true,
  "classificationId": "1",
  "classificationType": "Broad Listening"
},
{
  "id": "2",
  "title": "Videos",
  "isVisible": true,
  "classificationId": "1",
  "classificationType": "Broad Listening"
},
...
],
"meta": {
  "totalCount": 12
}
}
```

id - integer value indicating unique identifier of media type

title - string value indicating description of media type

isVisible - boolean value indicating visibility of media type in account

classificationId - integer value indicating unique identifier of media type classification

classificationType - string value indicating description of media type classification

Retrieve Media Type

This method retrieves a single media types based on the supplied ID.

```
/v3/mediaTypes/{id}
```

HTTP Method

```
GET
```

Request Parameters

mediaTypeId - integer value indicating media type to retrieve

Request Parameter Example

```
GET /v3/mediaTypes/{id}
```

Sample Request

```
GET /v3/mediaTypes/1
```

Sample Response

```
{
  "data": [
    {
      "id": "1",
      "title": "Blogs",
      "isVisible": true,
      "classificationId": "1",
      "classificationType": "Broad Listening"
    }
  ],
  "meta": {
    "totalCount": 1
  }
}
```

id - integer value indicating unique identifier of media type
title - string value indicating description of media type
isVisible - boolean value indicating visibility of media type in account
classificationId - integer value indicating unique identifier of media type classification
classificationType - string value indicating description of media type classification

CHAPTER 21 Posts

In this chapter ...

- [Retrieve Posts](#)

Use the Post endpoint to retrieve the social post content associated with a predefined topic profile. The Social Studio API allows an export of up to 1000 posts per topic profile every 30 seconds (equaling 2.8 million posts in 24 hours).



Note: The list of retrieved posts doesn't include posts processed by our privacy compliance systems. To include these posts, add the `&includeHiddenAuthorPosts=true` parameter to your request. The results include these posts, but the content is removed.

The Social Studio API enforces some content restriction due to legal requirements:

*The API removes all Twitter content and can only provide post and author IDs as well as workflow data for each post. We remove tweet content and post dynamics you can rehydrate directly via Twitter APIs. *We truncate any broad listening topic profile content to 310 characters or 10 percent of the content, whichever is greater.*

Retrieve Posts

```
GET /v3/posts
```

cURL Example

```
curl -X GET -H "access_token: 513e6576-218d-4a9e-b7cc-ac4ebf45fd1e" 'https://api.socialstudio.radian6.com/v3/posts?topics=1829'
```

Retrieve Posts

This method retrieves post data from available broad listening topic profiles and social accounts.

```
/v3/posts
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Request Parameter Sample

```
GET /v3/posts
```

Query Parameters

```
*topics* - string value containing comma-delimited list of topic profile IDs to include
in call (required)
*limit* - integer value indicating number of posts to return in the call (defaults to 25,
accepts values from 1 to 1000)
*sinceId* - integer value indicating ID of post to start retrieve. This parameter allows
only posts created after the identified post.
*beforeId* - integer value indicating ID of post to start retrieve. This parameter allows
only posts created before and including the identified post.
*startDate* - timestamp value in epoch milliseconds indicating the the start date for the
filter. This parameter allows only those posts published to the social media channel on
or after the given date, based on the *updated* attribute for the post (defaults to 24
hours before call occurs).
*endDate* - timestamp value in epoch milliseconds indicating the the end date for the
filter. This parameter allows only those posts published to the social media channel on
or before the given date, based on the *updated* attribute for the post (defaults to current
time).
*mediaTypes* - string value containing comma-delimited list of media type IDs to include
in call (defaults to all applicable media types including in the specified topic profiles)
*extendedMediaTypes* - string value containing comma-delimited list of extended media type
IDs to include in call (defaults to all applicable extended media types including in the
specified topic profiles)
*includeWorkflow* - boolean value indicates whether to include workflow with the returned
posts (defaults to false)
*includeSpam* - boolean value indicating whether to include posts flagged as spam by the
Salesforce Marketing Cloud (defaults to false)
*includeAnchors* - boolean value indicating whether to include anchor tags in content. A
*true* value indicates the call returns links as HTML anchor tags within the returned post
content. A *false* value returns plain text (defaults to false).
```

```

*includeAuditTrail* - true to get an array of workflow events performed on the post. This
  automatically sets the includeWorkflow flag to true
*highlight* - boolean value indicating whether the call highlights keywords. A *true* value
  indicating highlighted keywords from the topic profile configuration in the post content
  for use when presented in a web page. A *false* value returns plain text (defaults to
  false).
*keywords* - string value containing a comma-delimited list of keywords to use when returning
  only posts containing those values. This parameter includes up to 50 values and ignores
  duplicates.
*postLabel* - label you want to query for
*sortBy* - string values indicating how call sorts the response:
  *publishedDate* - sort by published date in descending order
  *publishedDate-ascending* - sort by published date in ascending order
  *blogPostId* - sort by post object ID in descending order (default)
  *blogPostId-ascending* - sort by post object ID in ascending order
  *rankBM25* - sort by relevance rank based on matched keyword frequency
*keywordGroups* - string value containing comma-delimited list of filter group IDs used
  to filter posts associated only with those groups. Defaults to all keyword groups provided
  in topic profiles from the *topics* parameter.
*includeSourceFilterMatches* - boolean value indicating whether to include the source
  groups containing URLs each post matches (defaults to false)
*regions* - string value containing comma-delimited list of region IDs (defaults to all
  regions assigned to topics from the *topic parameter)
*languages* - string value containing comma-delimited list of language IDs (defaults to
  all languages assigned to topics from the *topic parameter)
*assignedUser* - string value including a comma-delimited list of users IDs for users
  within the same client as the calling user
*engagement* - string value containing comma-delimited list of engagement status IDs to
  return
*priority* - integer value indicating priority values to use when returning posts:
  *1* - low
  *2* - medium
  *3* - high
*classification* - string value containing comma-delimited list of classifications to
  return
*eventsSince* - timestamp value used to indicate posts to return after provided value
  (milliseconds after midnight on 1/1/1970). Adding this parameter includes an HTTP header
  called Workflow-As-Of-Date that contains current timestamp of call. Use this value to
  retrieve only workflow events applied since previous method call. This parameter considers
  only workflow applied within last 30 days of call.
*unassigned* - boolean value indicating the call returns only unassigned posts
*sentiment* - string value containing comma-delimited list of sentiment IDs used to filter
  posts:
  *-10* - Negative
  *0* - Neutral
  *10* - Positive
*authorTags* - string value containing comma-delimited list of author tags used to filter
  posts

```

Sample Call

The call below returns posts from the specified topic profile:

```
GET /v3/posts?topics=1234
```

The call below returns the latest 50 Twitter posts:

```
GET /v3/posts?topics=1234&mediaTypes=8&limit=50
```

The call below returns the latest 50 Twitter posts since the specified ID

```
GET /v3/posts?topics=1234&mediaTypes=8&limit=50&sinceId=777777
```

Sample Response

```
{
  "data" : [
    {
      "id": "1828477438",
      "title": "Tennis Star visits Malawi",
      "externalId": null,
      "externalLink": "http://www.youtube.com/watch?v=exampleValue",
      "sourceApplication": null,
      "content": "",
      "summaryContent": "",
      "publishedDate": "2015-07-21T10:07:58Z",
      "harvestDate": "2015-07-21T11:23:23Z",
      "languageId": 1,
      "regionId": 249,
      "postStatus": 0,
      "mediaProvider": {
        "id": "2",
        "title": "YouTube",
        "mediaTypeId": 2,
        "extendedMediaType": null
      },
      "source": {
        "id": "2229562169",
        "title": "Tennis Star visits Malawi",
        "externalLink": "youtube.com",
        "language": "en",
        "region": null,
        "tags": [
          {
            "id": "7494",
            "value": "Sample source tag"
          }
        ]
      },
      "comments": [
        {
          "id": "56436",
          "value": "Sample source comment",
          "noteTypeId": "4"
        }
      ]
    },
    {
      "author": {
        "id": "-1",
        "title": "YouTube Channel",
```

```

    "externalId": "-1",
    "externalLegacyId": null,
    "authorFullName": "YouTube Channel",
    "avatar": null,
    "authorTags": [
      {
        "id": "5567567",
        "value": "Sample author tag"
      }
    ],
    "authorComments": [
      {
        "id": "556456456",
        "value": "Sample author comment"
      }
    ],
    "bio": null,
    "verified": false
  },
  "parent": null,
  "sentiment": null,
  "postStatusException": null,
  "engagement": 4564,
  "classification": 4,
  "tags": [
    {
      "id": "5466546",
      "value": "Sample tag"
    }
  ],
  "postDynamics": [
    {
      "fieldId": "5",
      "label": "View Count",
      "value": "0"
    },
    {
      "fieldId": "1",
      "label": "Comment Count",
      "value": "0"
    },
    {
      "fieldId": "2",
      "label": "Unique Commenters",
      "value": "0"
    },
    {
      "fieldId": "3",
      "label": "Engagement",
      "value": "0"
    },
    {
      "fieldId": "4",
      "label": "Likes and Votes",

```



```

        "value": "0"
      },
      {
        "fieldId": "7",
        "label": "Inbound Links",
        "value": "0"
      },
      {
        "fieldId": "21",
        "label": "Sentiment",
        "value": "0",
        "overridden": "false"
      }
    ],
    "metadata": null,
    "topics": [
      4526
    ],
    "sourceFilterMatches": null,
    "assignedUser": null,
    "comments": [
      {
        "id": "435345",
        "value": "Sample comment",
        "noteTypeId": "0"
      }
    ],
    "workflowHistory": [
      {
        "id": "790583",
        "workflowType": "BlogPostSentiment",
        "entityId": "2101931776",
        "creator": {
          "id": "2999",
          "displayName": "Bob",
          "email": "bob@radian6.com",
          "avatarUrl": null
        },
        "createdDate": "2015-11-24T20:35:02Z",
        "deletor": null,
        "deletedDate": null,
        "clientId": "-1",
        "value": "-10"
      }
    ]
  }
},
"meta" : {
  "totalCount" : 1
}

```

```

    }
  }

```

```

*id* - integer value indicating unique identifier for post object, used to assign and
access workflow associated with post
*title* - string value containing description of post
*externalId* - integer value indicating unique ID for post assigned by original social
media channel (such as Facebook or Twitter)
*externalLink* - string value containing URL of original post
*sourceApplication* - string value for name of application that created Twitter content
*content* - string value containing text of original post, including links
*summaryContent* - string value reserved for internal use (will always return empty)
*harvestDate* - timestamp value indicating when Social Studio discovered and indexed post
*languageId* - integer value indicating supported language for post (review Languages route
for available IDs)
*regionId* - integer value indicating region associated with post (review region route for
available IDs)
*postStatus* - integer value indicating post status:
    *0* - normal, active post
    *1* - spam (auto-detected)
    *2* - deleted or hidden post (based on user action)
*sentiment* - integer value indicating manually assigned sentiment value (requires
includeWorkflow=true)
*postStatusException* - integer value indicating manually assigned post status value
(requires includeWorkflow=true)
*engagement* - integer value indicating state(such as under review or closed) of post
(requires includeWorkflow=true)
*classification* - integer value indicating classification group assigned to post (requires
includeWorkflow=true)
*assignedUser* - integer value indicating user assigned to post (requires
includeWorkflow=true)
*priority* - integer value indicating priority assigned to post (requires
includeWorkflow=true)
*tags* - array of tag values attached to post (requires includeWorkflow=true)
*comments* - array of comment values attached to post (requires includeWorkflow=true)
*mediaProvider* - object containing information about post within context of original
social network:
    *id* - integer value indicating original social network
    *title* - string value indicating name of original social network
    *mediaTypeId* - number value indicating type of media
    *extendedMediaTypeId* - number value indicating type of extended media
*source* - object containing information about post source used to represent source in
Social Studio. Multiple posts may use the same source. Requires includeWorkflow=true.
    *id* - integer value indicating unique identifier of source
    *title* - string value containing text describing post
    *externalLink* - string value containing link to original post
    *language* - string value containing ISO language code
    *region* - string value containing region associated with post
    *tags* - array of string values indicating tags for post
    *verified* - boolean value indicating whether Twitter verified post
*author* - object containing information about individual or company that created post
(requires includeWorkflow=true)
    *id* - integer value indicating unique identifier for author

```

```
*title* - string value containing text describing author
*externalId* - integer value indicating unique identifier for author on social media
channel
*externalLegacyId* - integer value indicating unique legacy identifier for author on
social media channel
*authorFullName* - string value containing full name for author
*avatar* - string value containing location for avatar image file
*authorTags* - array containing ID and value pairs for tags attached to author
*authorComments* - array containing ID and value pairs for comments attached to author

*bio* - string value containing author bio information
*verified* - boolean value indicating whether Twitter verified author
*parent* - object containing IDs used by Social Studio and the social media channel to
identify parent post
*postDynamics* - object containing more information about the post (such as number of likes
or comments for a post). These numbers change over time and vary from social channel to
social channel.
*metadata* - object containing additional metadata about the post. This information can
change over time and vary from social channel to social channel.
*topics* - object containing information about the topics associated with the post (topics
assigned in request parameter)
*sourceFilterMatches* - object containing the source group with URLs the post matches
*parentAuthor* - object containing information on author of parent post for current post.
Currently used only for retweeted Twitter posts.
```

CHAPTER 22 Engage Macros

In this chapter ...

- [Retrieve Macros](#)
- [Retrieve Macro](#)
- [Create Macro](#)
- [Update Macro](#)
- [Execute Macro](#)
- [Delete Macro](#)
- [Retrieve SalesforceOrgs](#)
- [Retrieve SalesforceOrg](#)

This endpoint allows you to configure predefined metadata to automatically apply when performing a workflow on posts in Engage and Analyze scenarios. For example, you might want to apply a macro whenever you see a social post that mentions the brand in the following manner:

```
"@Brand, my laptop just crashed! Need help asap!"
```

The macro would execute the following actions:

```
*Add a post tag *Laptop*
*Set priority as *High*
*Send to Salesforce as a case
```

Create Macro

First, create a macro to define workflow tasks to execute:

```
POST /v3/macros
```

cURL Example

```
curl -X POST -H "access_token: 82200099-7799-4023-b76d-11bb7c941bf1"
-H "Content-Type: application/json" -d '{
  "name": "Urgent - Laptop Support Issue",
  "description": "Urgent Laptop Issues to be addressed ASAP",
  "actions": [
    {
      "type": "SetPriority",
      "value": "3"
    },
    {
      "type": "AddPostNote",
      "value": "Laptop"
    },
    {
      "type": "SendToSalesforceScs",
      "value": "00DD00000007aAJMAY",
      "parameters": [
        {
          "name": "case",
          "value": "on"
        }
      ]
    }
  ]
}
```

```
        {
          "name": "lead",
          "value": "off"
        }
      ]
    }
  ]
}' 'https://api.socialstudio.radian6.com/v3/macros'
```

Apply Macro

Once you create a macro, you can apply it to a post:

```
POST /v3/macros/{id}?action=execute
```

cURL Example

```
curl -X POST -H "Content-Type: application/json" -H "access_token:
82200099-7799-4023-b76d-11bb7c941bf1" -d '{
  "posts" : [1868932796],
  "topics" : [8]
}'
'https://api.socialstudio.radian6.com/v3/macros/23114?action=execute'
```

Retrieve Macros

This method retrieves all macros within a Social Studio organization. This macro can contain several different actions within the JSON payload, indicating actions to perform when the macro receives an execute call. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

```
/v3/macros
```

HTTP Method

```
GET
```

Query Parameters

```
*workspaceGroupID* - reserved for internal use  
*workspaceId* - integer value indicating return of only those macros from the specified workspace
```

Request Parameters Example

```
GET /v3/macros
```

Sample Request

```
GET /v3/macros
```

Sample Response

A successful call returns the following response (can be an array of macros:

```
"data": [  
  {  
    "id": 322,  
    "name": "Forward to Salesforce",  
    "description": "Sends post for further investigation",  
    "owner": {  
      "id": "1827",  
      "title": "John.Smith@domain.com",  
      "avatarURL": null  
    },  
    "client": 1,  
    "createdDate": "2014-03-26T16:04:29Z",  
    "modifiedDate": null,  
    "actions": [  
      {  
        "type": "SendToSalesforceScs",  
        "value": "00DR00000008ioUFGY",
```

```

        "parameters": [
            {
                "name": "case",
                "value": "on"
            },
            {
                "name": "lead",
                "value": "on"
            }
        ]
    }
]
},
],
"meta": {
    "totalCount": 1
}
}

```

id - integer value indicating unique ID for macro
 name - string value indicating name for macro
 description - string value including text describing the macro
 owner - integer value indicating unique ID for owner of macro
 client - integer value indicating unique ID for client containing macro (read-only)
 createdAt - timestamp value indicating when system created macro (read-only)
 modifiedDate - timestamp value indicating when system last updated macro (read-only)
 action - array of actions included within macro:

- *SetClassification* - integer value indicating media type classification to add
- *AddAuthorTag* - string value to add as tag to author object
- *AddPostTag* - string value to add as tag to post object
- *AssignUser* - integer value identifying post to assign to user
- *SetEngagement* - integer value identifying engagement level to set
- *SetPriority* - integer value identifying priority level to set
- *AddPostNote* - string value to add to post as note
- *SetSpamStatus* - integer value used to set spam status
 - *0* - not spam
 - *2* - spam
- *AddSourceTag* - string value to add as source tag
- *SetSentiment* - integer value to set as sentiment level
- *SendToSalesforceScs* - integer value used to set organization to which the macro will send post
- *case* - string value indicating whether to create case
 - *off* - do not create case
 - *on* - create case (default)
- *lead* - string value indicating whether to create lead
 - *off* - do not create lead (default)
 - *on* - create lead

Retrieve Macro

This method retrieves a macro within a Social Studio organization. This macro can contain several different actions within the JSON payload, indicating actions to perform when the macro receives an execute call. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

Resource

```
/v3/macros/{id}
```

HTTP Method

```
GET
```

Request Parameters

```
*id* - integer value indicating
```

Request Parameters Example

```
GET /v3/macros{id}
```

Sample Request

```
GET /v3/macros/1234
```

Sample Response

A successful call returns the following response (can be an array of macros:

```
{
  "data": [
    {
      "id": 322,
      "name": "Forward to Salesforce",
      "description": "Sends post for further investigation",
      "owner": {
        "id": "1827",
        "title": "John.Smith@domain.com",
        "avatarURL": null
      },
      "client": 1,
      "createdDate": "2014-03-26T16:04:29Z",
      "modifiedDate": null,
      "actions": [
        {
```



```

    "type": "SendToSalesforceScs",
    "value": "00DR00000008ioUFGY",
    "parameters": [
      {
        "name": "case",
        "value": "on"
      },
      {
        "name": "lead",
        "value": "on"
      }
    ]
  }
]
},
],
"meta": {
  "totalCount": 1
}
}

```

id - integer value indicating unique ID for macro
 name - string value indicating name for macro
 description - string value including text describing the macro
 owner - integer value indicating unique ID for owner of macro
 client - integer value indicating unique ID for client containing macro (read-only)
 createdAt - timestamp value indicating when system created macro (read-only)
 modifiedDate - timestamp value indicating when system last updated macro (read-only)
 action - array of actions included within macro:

- *SetClassification* - integer value indicating media type classification to add
- *AddAuthorTag* - string value to add as tag to author object
- *AddPostTag* - string value to add as tag to post object
- *AssignUser* - integer value identifying post to assign to user
- *SetEngagement* - integer value identifying engagement level to set
- *SetPriority* - integer value identifying priority level to set
- *AddPostNote* - string value to add to post as note
- *SetSpamStatus* - integer value used to set spam status
 - *0* - not spam
 - *2* - spam
- *AddSourceTag* - string value to add as source tag
- *SetSentiment* - integer value to set as sentiment level
- *SendToSalesforceScs* - integer value used to set organization to which the macro will send post
 - *case* - string value indicating whether to create case
 - *off* - do not create case
 - *on* - create case (default)
 - *lead* - string value indicating whether to create lead
 - *off* - do not create lead (default)
 - *on* - create lead

Create Macro

This method creates a macro within a Social Studio organization. This macro can contain several different actions within the JSON payload, indicating actions to perform when the macro receives an execute call. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

Resource

```
/v3/macros
```

HTTP Method

```
POST
```

Request Parameters

```
None
```

Request Parameters Example

```
POST /v3/macros
```

JSON Parameters

```
*name* - string value indicating name for macro (required)
*description* - string value including text describing the macro
*action* - array of actions included within macro:
    *SetClassification* - integer value indicating media type classification to add
    *AddAuthorTag* - string value to add as tag to author object
    *AddPostTag* - string value to add as tag to post object
    *AssignUser* - integer value identifying post to assign to user
    *SetEngagement* - integer value identifying engagement level to set
    *SetPriority* - integer value identifying priority level to set
    *AddPostNote* - string value to add to post as note
    *SetSpamStatus* - integer value used to set spam status
        *0* - not spam
        *2* - spam
    *AddSourceTag* - string value to add as source tag
    *SetSentiment* - integer value to set as sentiment level
    *SendToSalesforceScs* - integer value used to set organization to which the macro will
send post
    *case* - string value indicating whether to create case
        *off* - do not create case
        *on* - create case (default)
    *lead* - string value indicating whether to create lead
```

```
*off* - do not create lead (default)
*on* - create lead
```

Sample Request

```
POST /v3/macros
{
  "name" : "Add post tag",
  "description" : "Adds a tag to a post",
  "actions": [
    {
      "type": "AddPostTag",
      "value": "Post Tag"
    }
  ]
}
```

Sample Response

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "23719"
    }
  ],
  "meta": {
    "totalCount": 1
  }
}
```

```
*id* - integer value indicating unique ID for macro
*owner* - integer value indicating unique ID for owner of macro
*client* - integer value indicating unique ID for client containing macro (read-only)
*createdDate* - timestamp value indicating when system created macro (read-only)
*modifiedDate* - timestamp value indicating when system last updated macro (read-only)
```

Update Macro

This method updates a macro within a Social Studio organization. This macro can contain several different actions within the JSON payload, indicating actions to perform when the macro receives an execute call. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

Resource

```
/v3/macros/{id}
```

HTTP Method

```
PUT
```

Request Parameters

```
*id* - integer value indicating unique ID for macro
```

Request Parameters Example

```
PUT /v3/macros
```

JSON Parameters

```
*name* - string value indicating name for macro (required)
*description* - string value including text describing the macro
*actions* - array of actions included within macros:
*action* - array of actions included within macro:
    *SetClassification* - integer value indicating media type classification to add
    *AddAuthorTag* - string value to add as tag to author object
    *AddPostTag* - string value to add as tag to post object
    *AssignUser* - integer value identifying post to assign to user
    *SetEngagement* - integer value identifying engagement level to set
    *SetPriority* - integer value identifying priority level to set
    *AddPostNote* - string value to add to post as note
    *SetSpamStatus* - integer value used to set spam status
        *0* - not spam
        *2* - spam
    *AddSourceTag* - string value to add as source tag
    *SetSentiment* - integer value to set as sentiment level
    *SendToSalesforceScs* - integer value used to set organization to which the macro will
    send post
    *case* - string value indicating whether to create case
        *off* - do not create case
        *on* - create case (default)
    *lead* - string value indicating whether to create lead
        *off* - do not create lead (default)
        *on* - create lead
```

Sample Request

```
PUT /v3/macros/1234
{
  "name": "My Macro",
  "description": "Macro Description",
  "actions": [
    {
```

```
        "type": "AddPostTag",
        "value": "Post Tag"
      }
    ]
  }
```

Sample Response

A successful call returns HTTP 200

Execute Macro

This method executes a macro within a Social Studio organization. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

Resource

```
/v3/macros/{id}
```

HTTP Method

```
POST
```

Request Parameters

id - integer value indicating unique ID for macro

Query Parameters

action - string value indicating action to perform
execute

JSON Parameters

posts - array of integer values identifying posts included in the running macro
topics - array of integer values identifying topics included in the running macro

Request Parameters Example

```
POST /v3/macros/{id}
```

Sample Request

```
POST /v3/macros/{id}?action=execute
{
  "posts": [2345],
  "topics" : [8]
}
```

Sample Reponse

A successful call returns the following response:

```
{
  "results": [
    {
      "postId": <2345>,
      "action": "AddPostTag",
      "result": "SUCCESS",
      "error": ""
    }
  ]
}
```

```
*results* - object including data on macro execution
*postId* - integer value indicating post on which macro performed
*action* - array of actions included within macro:
  *SetClassification* - integer value indicating media type classification to add
  *AddAuthorTag* - string value to add as tag to author object
  *AddPostTag* - string value to add as tag to post object
  *AssignUser* - integer value identifying post to assign to user
  *SetEngagement* - integer value identifying engagement level to set
  *SetPriority* - integer value identifying priority level to set
  *AddPostNote* - string value to add to post as note
  *SetSpamStatus* - integer value used to set spam status
    *0* - not spam
    *2* - spam
  *AddSourceTag* - string value to add as source tag
  *SetSentiment* - integer value to set as sentiment level
  *SendToSalesforceScs* - integer value used to set organization to which the macro will
  send post
  *case* - string value indicating whether to create case
    *off* - do not create case
    *on* - create case (default)
  *lead* - string value indicating whether to create lead
    *off* - do not create lead (default)
    *on* - create lead
```

Delete Macro

This method deletes a macro within a Social Studio organization. Use the Salesforce Organization methods to retrieve necessary Service Cloud values for this call.

Resource

```
/v3/macros/{id}
```

HTTP Method

```
DELETE
```

Request Parameters

id - integer value indicating unique ID for macro

Request Parameters Example

```
DELETE /v3/macros/{id}
```

Sample Request

```
DELETE /v3/macros/{id}
```

Sample Reponse

A successful call returns a No Content 204 code, indicating the call succeeded and no content exists.

A call that matches no macro ID value returns a 404 Not Found code.

Retrieve SalesforceOrgs

This method returns information on all Salesforce organizations registered within the tenant for the user executing the call.

```
/v3/salesforceOrganizations
```

HTTP Method

```
GET
```

Request Parameters

```
None
```

Request Parameters Example

```
GET /v3/salesforceOrganizations
```

Sample Request

```
GET /v3/salesforceOrganizations
```

Sample Request

A successful call returns the following response:

```
{
  "data": [
    {
      "id": "00D61000000YMhPABC",
      "title": "prod.r64sf",
      "isSCS": false,
      "isSandbox": false,
      "isPersonAccountCapable": false,
      "adminEmail": "john.doe@salesforce.com",
      "integrationUserId": 6110,
      "tenantId": 1
    }
  ],
  "meta" : {
    "totalCount" : 1
  }
}
```

id - string value containing the 18-character unique ID for the Salesforce organization
 link - string value containing the URL location for the organization
 updated - timestamp value indicating when the last organization update occurred
 title - string value indicating the name of the organization
 isSCS - boolean value indicating whether the organization uses the Social Cloud
 isSandbox - boolean value indicating whether the organization functions as a developer sandbox
 isPersonAccountCapable - boolean value indicating whether the organization can use person accounts
 adminEmail - string value including the email address for the organization administrator
 integrationUserId - integer value indicating the unique ID for the organization integration user
 tenantId - integer value indicating the unique ID for the tenant associated with the Salesforce organization

Retrieve SalesforceOrg

This method returns information on a specified Salesforce organization registered within the tenant for the user executing the call.

```
/v3/salesforceOrganizations/{id}
```


HTTP Method

```
GET
```

Request Parameters

```
*id* - string value containing the 18-character unique ID for the Salesforce organization (required)
```

Request Parameters Example

```
GET /v3/salesforceOrganizations/{id}
```

Sample Request

```
GET /v3/salesforceOrganizations/00DA00000000000001
```

Sample Request

A successful call returns the following response:

```
{
  "id": "00DA00000000000001",
  "link":
"https://stg3-api.dev.ca1.sfmt.co/socialcloud/v3/salesforceOrganizations/00DA00000000000001",
  "updated": "2015-01-01T00:00:00Z",
  "title": "Organization Name",
  "isSCS": true,
  "isSandbox": false,
  "isPersonAccountCapable": false,
  "adminEmail": "orgowner@example.com",
  "integrationUserId": 10010010,
  "tenantId": 1
}
```

```
*id* - string value containing the 18-character unique ID for the Salesforce organization
*link* - string value containing the URL location for the organization
*updated* - timestamp value indicating when the last organization update occurred
*title* - string value indicating the name of the organization
*isSCS* - boolean value indicating whether the organization connects to the Service Cloud via Social Customer Service integration
*isSandbox* - boolean value indicating whether the organization functions as a developer sandbox
*isPersonAccountCapable* - boolean value indicating whether the organization can use person accounts
*adminEmail* - string value including the email address for the organization administrator
*integrationUserId* - integer value indicating the unique ID for the organization integration
```

```
user
*tenantId* - integer value indicating the unique ID for the Social Studio tenant associated
with the Salesforce organization
```

CHAPTER 23 Radian6 API Appendix

In this chapter ...

- [Supported Browsers](#)
- [Supported Editions](#)
- [Quick Start](#)
- [Using the Services](#)
- [Resources](#)

If you are using the older Radian6 API, use this appendix to extend the functionality of Marketing Cloud.

The Radian6 API extends the functionality of Marketing Cloud in these ways.

- Create custom reporting and make your own visualizations.
- Extract post data directly from your Topic Profiles.
- Extract the data from visualizations or widgets from the Radian6 Dashboard.
- Access appended post data like post tags and source tags.
- Access Radian6 Insights data like demographics, sentiment, and entities, and so on.

Use this data to drive integration, from creating custom internal reporting to enhancing the value of your applications and services.

Supported Browsers

Radian6 supports the following browsers:

- Mozilla® Firefox®
- Google Chrome™
- Microsoft® Internet Explorer®
- Apple® Safari®

Older versions of Internet Explorer might not support the Summary Dashboard. We recommend Internet Explorer 9 and above, or another browser listed above.

Supported Editions

Radian6 supports these Editions:

- Developer Edition
- Enterprise Edition
- Unlimited Edition

If you're an existing customer and want to upgrade to any of these editions, contact your account representative.

Quick Start

The Radian6 REST API retrieves, analyzes, and modifies social media posts and data from any Topic profile in your Radian6 account.

Before you begin, make sure that you have:

- A valid username and password
- A unique application key for your application

If you don't already have an application key, contact your account representative.

The following endpoints are available for access to the Radian6 API.

Development endpoint

`https://demo-api.radian6.com/socialcloud/v1/`

All development work should be performed and tested on the development endpoint before running your application against production.

Production endpoint

`https://api.radian6.com/socialcloud/v1/`

Run your application against the production endpoint after you have tested it on the development endpoint.

If you currently have an account on only one of the two environments, please contact apisupport@radian6.com in order to have your account replicated and available at both endpoints.

To get started with the Radian6 API, follow these steps.

1. Authenticate with the API.
2. Issue a basic call to one of the API methods.
3. Fetch some data from the Radian6 system.

[Step One: Authenticate with the API](#)

You must authenticate with the API before issuing calls.

[Step Two: Issue a Call to a Method](#)

After authenticating with the Radian6 API, issue a call to an API method.

[Step Three: Fetch Data](#)

After authenticating with the Radian6 API and getting a list of topics, you can now fetch some data for a topic.

Step One: Authenticate with the API

You must authenticate with the API before issuing calls.

The API currently only supports a basic authentication mechanism. An initial call needs to be made to the authentication service in order to fetch an authentication token used in subsequent API calls.

```
https://api-endpoint/auth/authenticate
```

This call expects the following request headers for an authentication check.

Parameter	Description
auth_user	The username on your account
auth_pass	The password in plain text
auth_appkey	The API key to include in your header parameter

All parameters are required. If authentication was successful, XML containing basic user account information and a token tag will be returned in the following format:

```
<token>70d756801c703f3e78f81726c11b00249fb81770a446958b2577cd223811e</token>
```

This is the token used to perform subsequent API requests.

```
<auth>
<token>e008252b4dce9b29c4c8155f0010cc8e128290b9e3ae99c8e9d15c
</token>
<UserDetails>
  <user>
    <userId>132972</userId>
    <clientId>1226</clientId>
    <displayName>Joe User</displayName>
    <emailAddress>joe.user@radian6.com</emailAddress>
    <packages />
  </user>
  <Packages>
    <feature>
      <featureId>1</featureId>
      <description>Workflow</description>
    </feature>
    <feature>
      <featureId>4</featureId>
      <description>Admin Portal Full</description>
    </feature>
  </Packages>
</UserDetails>
```

```

</feature>
<feature>
  <featureId>6</featureId>
  <description>Require PO Number</description>
</feature>
<feature>
  <featureId>8</featureId>
  <description>SENTIMENT</description>
</feature>
</Packages>
</UserDetails>
</auth>

```

Step Two: Issue a Call to a Method

After authenticating with the Radian6 API, issue a call to an API method.

Now that you have the authentication token and API application key, you can make a call to one of the API methods. Fetching data requires the use of Topic Profiles, so let's start by getting a list of Topics. The call used to fetch the list of topics is `TopicService.fetchTopicList`.

```
https://api-endpoint/topics
```

As with all other calls, you must provide two request headers, `auth_token` and `auth_appkey`, obtained in [Step One: Authenticate with the API](#).

You should receive an XML response containing a list of Topic Profiles and related information.

Sample XML Response Format

```

<topicFilters>
  <topicFilter>
    <name><![CDATA[2013 Candidates]]></name>
    <public>1</public>
    <status>1</status>
    <estimateVolume>317900</estimateVolume>
    <competeEnabled>0</competeEnabled>
    <bCode></bCode>
    <creatorId>2</creatorId>
    <creatorName>Chris Doe</creatorName>
    <creatorEmail>someguy@radian6.com</creatorEmail>
    <createDate>20080124</createDate>
    <topicFilterId>232</topicFilterId>
    <sentiment>0</sentiment>
    <postTopicIgnoreStatus>-1</postTopicIgnoreStatus>
    <inboundOnTopicLinksCount>0</inboundOnTopicLinksCount>
    <number_queries>13</number_queries>
    <filterGroups>
      <filterGroup>
        <filterGroupId>541</filterGroupId>
        <name><![CDATA[Group 1]]></name>
        <filterGroupId>1</filterGroupId>
        <filterQueries>

```

```

        <filterQuery>
          <query>"John" AND "president"</query>
          <filterQueryId>2031</filterQueryId>
          <isExcludeQuery>>false</isExcludeQuery>
        </filterQuery>
        <filterQuery>
          <query>"Jane" AND "ceo"</query>
          <filterQueryId>2039</filterQueryId>
          <isExcludeQuery>>false</isExcludeQuery>
        </filterQuery>
      </filterQueries>
    </filterGroup>
  </filterGroups>
  <sentimentQueries></sentimentQueries>
  <includeSourceFilterList>
    <filterIds></filterIds>
  </includeSourceFilterList>
  <excludeSourceFilterList>
    <filterIds></filterIds>
  </excludeSourceFilterList>
  <includeAllSourceFilterList>
    <filterIds></filterIds>
  </includeAllSourceFilterList>
  <languages></languages>
  <mediaType></mediaType>
  <projects></projects>
  <regions></regions>
</topicFilter>
</topicFilters>

```

Step Three: Fetch Data

After authenticating with the Radian6 API and getting a list of topics, you can now fetch some data for a topic.

This example shows you how to fetch the most recent 100 items from the Radian6 system for a topic over the last 24 hours. The call used to fetch this data is `DataService.fetchRecentTopicPosts`.

```
https://api-endpoint/data/topicdata/recent/{recentXhours}/{topics}
/{mediatypes}/{PageIndex}/{pageSize}
```

recentXHours

24 (the last 24 hours of data)

topics

232 (a topicId fetched from the call in [Step Two: Issue a Call to a Method](#))

mediatypes

1,2,3,4 (fetched from `LookupService.fetchMediaTypes`)

PageIndex

1 (the first page)

pageSize

100 (all items on current page)

For example, the call to development endpoint would look like this:

```
https://sandbox-insights.radian6.com/socialcloud/v1/data/topicdata
/realtime/24/232/1,2,3,4/1/100
```

This call returns a list of matching posts, sorted by default sort by `publish_date`. Each item in the result set is defined by an `<Article>` tag.

```
<radian6_RiverOfNews_export>
  <report_date>Fri Oct 30 10:22:06 ADT 2009</report_date>
  <user_name>Jane Smith</user_name>
  <RoN_sort_order>publishedDate</RoN_sort_order>
  <article_count>1</article_count>
  <article ID="1934621185">
    <description charset="UTF-8">
      <headline><![CDATA[ TWEET FROM: ACME]]>
      </headline>
      <author><![CDATA[ ACME]]>
      </author>
      <content><![CDATA[ The content of the Tweet]]>
      </content>
    </description>
    <source><![CDATA[ TWEET FROM: ACME]]></source>
    <host><![CDATA[ twitter.com]]></host>
    <article_url> <![CDATA[ http://twitter.com/username/statuses/4735539663]]>
    </article_url>
    <media_provider>TWITTER</media_provider>
    <media_type_id>8</media_type_id>
    <spam_rating>TODO</spam_rating>
    <publish_date>Oct 09, 2009 11:31 AM</publish_date>
    <harvest_date>Oct 09, 2009 11:31 AM</harvest_date>
    <PostDynamicsIteration>
      <PostDynamicsDefinition>
        <fieldId>9</fieldId>
        <label>Following</label>
        <value>0</value>
        <sortOrder>1</sortOrder>
      </PostDynamicsDefinition>
      <PostDynamicsDefinition>
        <fieldId>8</fieldId>
        <label>Followers</label>
        <value>0</value>
        <sortOrder>2</sortOrder>
      </PostDynamicsDefinition>
      <PostDynamicsDefinition>
        <fieldId>10</fieldId>
        <label>Updates</label>
        <value>0</value>
        <sortOrder>3</sortOrder>
      </PostDynamicsDefinition>
      <PostDynamicsDefinition>
        <fieldId>21</fieldId>
        <label>Sentiment</label>
        <shortLabel>S</shortLabel>
```



```

        <sortOrder>4</sortOrder>
        <value>15418,0</value>
        <exceptionValue>15418,false</exceptionValue>
        <reportValue>Neutral</reportValue>
        <tooltip />
    </PostDynamicsDefinition>
    <reportFormattedData><![CDATA[ <span style="font-weight:bold; color: #FF9900;
font-size: 11pt"> Following: </span>0 <span style="font-weight:bold; color: #FF9900;
font-size: 11pt"> Followers:
    </span>0 <span style="font-weight:bold; color: #FF9900; font-size: 11pt"> Updates:
</span>0 <span style="font-weight:bold; color: #FF9900; font-size: 11pt"> Sentiment:
</span>Neutral  ]]>
    </reportFormattedData>
  </PostDynamicsIteration>
</article>
</radian6_RiverOfNews_export>

```

Using the Services

Access and manage your posts, users, insights, topics, and other data.

This section walks you through common operations for each of the services.

[Post Service](#)

The Post Service enables you to perform operations such as assigning users to posts, setting the engagement type on posts, and adding tags to posts.

[User Service](#)

The User Service enables you to perform operations such as retrieving user details and their dashboards.

[Insight Service](#)

The Insight Service enables you to perform operations such as aggregating insights and returning the insight types.

[Topic Service](#)

The Insight Service enables you to perform operations such as creating topic profiles, filter groups, and filter queries.

[Data Service](#)

The Data Service enables you to perform operations such as fetching posts and topic comparison data.

[Blog Service](#)

The Blog Service enables you to perform operations such as fetching a list of posts for a given site and adding notes to a site.

[Authentication Service](#)

The Authentication Service enables you to authenticate a user in the Radian6 system.

[Lookup Service](#)

The Lookup Service enables you to perform operations such as fetching a list of media types, languages, users, and workflow items.

Post Service

The Post Service enables you to perform operations such as assigning users to posts, setting the engagement type on posts, and adding tags to posts.

Calls can be made no more than once every 30 seconds.

Get Post Details

Fetch post details, such as the content, title, author, and published date.

```
GET /post
```

Parameters	Type	Description
auth_token	HeaderParam	Required. Request header containing the token returned from authentication with the API.
auth_appkey	HeaderParam	Required. Application key unique to your account.
url	PathParam	URL of the post.

Example

```
http://api.radian6.com/socialcloud/v1/post?url=http://twitter.com/
username/statuses/13...4
```

Request Headers

```
GET /socialcloud/v1/post?url HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<PostDetails>
  <blogPost>
    <Post>
      <postId>12...</postId>
      <title>
        <![CDATA[Tweet from username (r6ts) ]]>
      </title>
      <author>
        <![CDATA[username]]>
      </author>
      <content>
```

```

        <![CDATA[]]>
    </content>
    <publishedDate>1321898901000</publishedDate>
    <link>
        <![CDATA[http://twitter.com/username/statuses/13.....3]]>
    </link>
    <providerId>10</providerId>
    <mediaTypeId>1</mediaTypeId>
    <languageId>1</languageId>
    <regionId>235</regionId>
    <postStatusId>0</postStatusId>
</Post>
</blogPost>
<blog>
    <blogId>44....7</blogId>
    <title>
        <![CDATA[Twitter / username]]>
    </title>
    <feed>
        <![CDATA[http://twitter.com/statuses/user_timeline/username.atom]]>
    </feed>
    <link>
        <![CDATA[http://twitter.com/username]]>
    </link>
    <languageId>1</languageId>
    <languageAccuracy>0.99</languageAccuracy>
</blog>
</PostDetails>
```

[Resources for Post Service](#)

The following list shows other operations you can use with the Post Service.

Resources for Post Service

The following list shows other operations you can use with the Post Service.

Operations	Example
Assign user to post	<pre>POST /post/workflow/assign/{postId}/{userId} /{topicList}</pre>
Assign post classification	<pre>POST /post/workflow/classification/{postId} /{classificationTypeId}</pre>
Set post engagement	<pre>POST /post/workflow/engagement/{postId} /{engagementTypeId}</pre>

Operations	Example
Assign post sentiment	<pre>POST /post/workflow/sentiment/{postId}/{TopicId} /{sentimentValue}</pre>
Add post note	<pre>POST /post/workflow/note/{postId}</pre>
Add post note reply	<pre>POST /post/workflow/notereply/{postId}</pre>
Add tags to posts	<pre>POST /post/workflow/tags/{postId}</pre>
Toggle spam	<pre>POST /post/workflow/toggleSpam/{postList} /{topicList}/{spamValue}</pre>
Get post dynamics	<pre>GET /post/metrics/{postId}</pre>
Get post workflow updates	<pre>GET /post/workflow/updates/{epoch}/{postIdList}</pre>
Remove tags and notes	<pre>POST /post/workflow/removeTagsAndNotes /{tagAndNoteIds}</pre>
Get child posts count	<pre>GET /v1/post/{parentPostId}/childcount</pre>
Get child posts count for multiple posts	<pre>GET /v1/post/list/{parentPostIdList}/childcount</pre>

The `auth_token` and `auth_appkey` header parameters are required for all calls.

User Service

The User Service enables you to perform operations such as retrieving user details and their dashboards.

Calls can be made no more than once every 30 seconds.

Get User

Return basic information for the user making the request.

```
GET /user
```

Parameters	Type	Description
auth_token	HeaderParam	Required. Request header containing the token returned from authentication with the API.
auth_appkey	HeaderParam	Required. Application key unique to your account.

Example

```
http://api.radian6.com/socialcloud/v1/user
```

Request Headers

```
GET /socialcloud/v1/smm/user HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<user>
  <userId>538</userId>
  <clientId>1</clientId>
  <displayName>
    <![CDATA[John Doe]]>
  </displayName>
  <emailAddress>John.Doe@example.com</emailAddress>
  <timezone>GMT</timezone>
</user>
```

[Resources for User Service](#)

The following list shows other operations you can use with the User Service.

Resources for User Service

The following list shows other operations you can use with the User Service.

Operations	Example
Get user details	GET /socialcloud/v1/user/details
Get client	GET /client
Get dashboard widgets	GET /user/dashboard
Set avatar	GET /user/avatar

The `auth_token` and `auth_appkey` header parameters are required for all calls.

Insight Service

The Insight Service enables you to perform operations such as aggregating insights and returning the insight types.

Get Insight Types

Fetch a list of the associated Insight Types for an Insights Package. The client must have access to the provided Topic Profile, and the Topic Profile must be subscribed to the provided Insights Package.

```
GET /socialcloud/v1/insights/insightTypes
```

Parameters	Type	Description
<code>auth_token</code>	HeaderParam	Required. Request header containing the token returned from authentication with the API.
<code>auth_appkey</code>	HeaderParam	Required. Application key unique to your account.
<code>packageId</code>	QueryParam	The ID of the Insights Package for which to fetch a list of Insight Types.
<code>topicProfileId</code>	QueryParam	The ID of the Topic Profile that is subscribed to the provided Insights Package.

Example

```
http://api.radian6.com/socialcloud/v1/insights/insightTypes
```

Request Headers

```
GET /socialcloud/v1/insights/insightTypes?packageId=1...2&topicProfileId=1...2 HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Mon, 05 Dec 2011 14:24:31 GMT
Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 135
```

Response

```
<insightTypes>
  <insightType>
    <dataDescriptor>multi-value</dataDescriptor>
    <description>Retweeted usernames</description>
    <displayName>Retweeted Usernames</displayName>
    <isPrivacyRelated>false</isPrivacyRelated>
    <name>retweet_username</name>
    <objectId>4d6...bbb</objectId>
    <providerName>radian6</providerName>
  </insightType>
  ...
</insightTypes>
```

[Resources for Insight Service](#)

The following list shows other operations you can use with the Insight Service.

Resources for Insight Service

The following list shows other operations you can use with the Insight Service.

Operations	Example
Aggregate insights	<pre>POST /socialcloud/v1/insights /aggregateInsightsByTopic</pre>
Get package subscriptions	<pre>GET /socialcloud/v1/insights/packageSubscriptions</pre>
Filter values	<pre>GET /socialcloud/v1/insights/fetchFilterValues /{topicFilterId}</pre>
Source insights	<pre>GET /socialcloud/v1/insights/fetchInsightsBySource /{topicProfileId}/{blogIds}/{providers}</pre>

The `auth_token` and `auth_appkey` header parameters are required for all calls.

Topic Service

The Insight Service enables you to perform operations such as creating topic profiles, filter groups, and filter queries.

Create or Update Topic Profile

Create or update a topic.

```
POST /topics/createTP/{topicId}/{name}/{isPublic}/{mediatypes}
/{languages}/{regions}
```

Parameters	Type	Description
auth_token	HeaderParam	Required. Request header containing the token returned from authentication with the API.
auth_appkey	HeaderParam	Required. Application key unique to your account.
topicId	PathParam	Unique Id of topic (required when updating)
name	PathParam	Display name of the topic filter
isPublic	PathParam	Integer value indicating if the topic is public or private
mediatypes	PathParam	Comma delimited list of valid media types for the topic
languages	PathParam	Comma delimited list of valid languages for the topic
regions	PathParam	Comma delimited list of valid regions for the topic
billingCode	QueryParam	The billing code for the topic. Default is an empty string.
isTrial	QueryParam	Integer value indicating if the topic is a trial. Default is 1.

Example

```
http://api.radian6.com/socialcloud/v1/topics/createTP/1/My Topic/1/8/1/2
```

Request Headers

```
GET /socialcloud/v1/topics/createTP/{topicId}/{name}/{isPublic}
/{mediatypes}/{languages}/{regions} HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
```



```
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<topicFilter>
  <name>
    <![CDATA[My Topic]]>
  </name>
  <public>0</public>
  <status>2</status>
  <estimateVolume>-1</estimateVolume>
  <competeEnabled>0</competeEnabled>
  <topicFilterTypeId>1</topicFilterTypeId>
  <bCode>
    <![CDATA[]]>
  </bCode>
  <creatorId>538</creatorId>
  <creatorName>
    <![CDATA[Jane Smith]]>
  </creatorName>
  <creatorEmail>Jane.Smith@...</creatorEmail>
  <topicFilterId>3...3</topicFilterId>
  <inboundOnTopicLinksCount>0</inboundOnTopicLinksCount>
  <languages/>
  <mediaType>8,12,14,13,11,10,9,5,2,1,16,4</mediaType>
  <projects/>
  <regions/>
  <deactivationDate>null</deactivationDate>
  <evp>false</evp>
  <topicFilterTier>
    <topicFilterTierId>-1</topicFilterTierId>
    <name/>
    <lowerTrafficLimit>-1</lowerTrafficLimit>
    <upperTrafficLimit>-1</upperTrafficLimit>
  </topicFilterTier>
  <number_queries>0</number_queries>
  <filterGroups>
    <filterGroup>
      <filterGroupId>2...9</filterGroupId>
      <name>
        <![CDATA[Group 1]]>
      </name>
      <filterGroupId>1</filterGroupId>
      <filterQueries/>
    </filterGroup>
  </filterGroups>
  <sentimentQueries/>
  <includeSourceFilterList>
    <filterIds/>
  </includeSourceFilterList>
  <excludeSourceFilterList>
    <filterIds/>
```

```

    </excludeSourceFilterList>
    <includeAllSourceFilterList>
      <filterIds/>
    </includeAllSourceFilterList>
  </topicFilter>

```

[Resources for Topic Service](#)

The following list shows other operations you can use with the Topic Service.

Resources for Topic Service

The following list shows other operations you can use with the Topic Service.

Operations	Example
Get multiple topic profiles	GET /topics
Get topic profile	GET /topics/{topicId}
Get topic profile usage	GET /topics/usage
Delete topic profile	POST /topics/remove/{topicId}
Create source filter association	POST /topics/{topicId}/sourcefilters
Get multiple filter groups	GET /topics/{topicId}/filterGroups
Get filter group	GET /topics/{topicId}/filterGroups/{filterGroupId}
Get source filters	GET /topics/{topicId}/sourcefilters
Delete filter group	DELETE /topics/{topicId}/filterGroups/{filterGroupId}
Create filter group	POST /topics/{topicId}/filterGroups
Update filter group	PUT /topics/{topicId}/filterGroups/{filterGroupId}
Get multiple filter queries	GET /topics/{topicId}/filterGroups/{filterGroupId}/filterQueries
Create filter query	GET /topics/{topicId}/filterGroups/{filterGroupId}/filterQueries
Get filter query	GET /topics/{topicId}/filterGroups/{filterGroupId}/filterQueries/{filterQueryId}
Delete filter query	DELETE /topics/{topicId}/filterGroups/{filterGroupId}/filterQueries/{filterQueryId}

Operations**Example**

Get insight subscription window GET /topics/{topicId}/subscriptionWindow

The auth_token and auth_appkey header parameters are required for all calls.

Data Service

The Data Service enables you to perform operations such as fetching posts and topic comparison data.

Get Post Data

Fetch posts matching given query parameters.

```
GET /data/topicdata/realtime/{recentXhours}/{topics}/{mediatypes}
/{pageIndex}/{pageSize}
```

Parameters	Type	Description
auth_token	HeaderParam	Required. Request header containing the token returned from authentication with the API.
auth_appkey	HeaderParam	Required. Application key unique to your account.
recentXhours	PathParam	Number of hours to go back. For example, setting this to 48 will return all posts within the last two days. This references the published date as they are returned within the response.
topics	PathParam	Comma delimited list of topic profile ids to get posts for.
mediaTypes	PathParam	Comma delimited list of media types from which post will be returned.
pageIndex	PathParam	Specifies which page of data to return.
pageSize	PathParam	Number of posts to return per page.

Example

```
http://api.radian6.com/socialcloud/v1/data/topicdata/realtime
/2/3..9/1,2,4,5,8,10,9,11,12,13,14,16/1/20?includeWorkflow=1&includeSpam=0
&merged=1&token=1321987774350&extendedMediaTypes=2,3,4
```

Request Headers

```
GET /socialcloud/v1/data/topicdata/realtime/{recentXhours}/{topics}
/{mediatypes}/{pageIndex}/{pageSize} HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<radian6_RiverOfNews_export>
  <report_date>2012-07-11 02:24:04 +1200</report_date>
  <user_name>Jane.Smith@...</user_name>
  <RoN_sort_order>publishedDate</RoN_sort_order>
  <article_count>2</article_count>
  <total_article_count>101</total_article_count>
  <article ID="2...9">
    <description charset="UTF-8">
      <headline/>
      <author fbid="-1" externalId="3...2"/>
      <author_full_name />
      <recipient/>
      <content/>
      <external_id>2...73</external_id>
      <parentExternalId>2...70</parentExternalId>
    </description>
    <avatar/>
    <source/>
    <host>
      <![CDATA[twitter.com]]>
    </host>
    <article_url>
      <![CDATA[2...73]]>
    </article_url>
    <media_provider>TWITTER</media_provider>
    <media_type_id>8</media_type_id>
    <language_id>16</language_id>
    <spam_rating>0</spam_rating>
    <publish_date epoch="1341901361000">2012-07-10 18:22:41 +1200</publish_date>
    <harvest_date epoch="1341901379000">2012-07-10 18:22:59 +1200</harvest_date>
    <PostInsights>
      <PostInsight>
        <Provider>
          <![CDATA[provider_name]]>
        </Provider>
        <Type>
          <![CDATA[type_name]]>
        </Type>
        <Value>
          <![CDATA[some_value]]>
        </Value>
      </PostInsight>
    </PostInsights>
  </article>
</radian6_RiverOfNews_export>
```

```

        </PostInsight>
    </PostInsights>
    <SourceInsights>
        <SourceInsight>
            <Provider>
                <![CDATA[provider_name]]>
            </Provider>
            <Type>
                <![CDATA[type_name]]>
            </Type>
            <Value>
                <![CDATA[some_value]]>
            </Value>
        </SourceInsight>
    </SourceInsights>
    <PostDynamicsIteration>
        <PostDynamicsDefinition>
            <fieldId>9</fieldId>
            <label>Following</label>
            <value/>
            <sortOrder>1</sortOrder>
        </PostDynamicsDefinition>
        <PostDynamicsDefinition>
            <fieldId>8</fieldId>
            <label>Followers</label>
            <value/>
            <sortOrder>2</sortOrder>
        </PostDynamicsDefinition>
        <PostDynamicsDefinition>
            <fieldId>10</fieldId>
            <label>Updates</label>
            <value/>
            <sortOrder>3</sortOrder>
        </PostDynamicsDefinition>
        <PostDynamicsDefinition>
            <fieldId>21</fieldId>
            <label>Sentiment</label>
            <shortLabel>S</shortLabel>
            <sortOrder>4</sortOrder>
            <value/>
            <exceptionValue>2860,false</exceptionValue>
            <reportValue>Neutral</reportValue>
            <tooltip/>
        </PostDynamicsDefinition>
        <reportFormattedData/>
    </PostDynamicsIteration>
</article>
<article ID="252359343">
    <description charset="UTF-8">
        <headline>
            <![CDATA[Post from Facebook user]]>
        </headline>
        <author fbid="1769972299" externalId="1769972299">
            <![CDATA[Facebook user]]>
        </author>
    </description>
</article>

```

```

    </author>
    <recipient>
      <![CDATA[None]]>
    </recipient>
    <content>
      <![CDATA[Facebook post content...]]>
    </content>
    <external_id>17...9_22...4</external_id>
  </description>
  <source ID="1...2">
    <![CDATA[Post from Facebook user]]>
  </source>
  <host>
    <![CDATA[www.facebook.com]]>
  </host>
  <article_url>
    <![CDATA[http://www.facebook.com/permalink.php?story_fbid=22...4&id=17...9]]>
  </article_url>
  <media_provider>facebook.com Discussions</media_provider>
  <media_type_id>12</media_type_id>
  <language_id>1</language_id>
  <spam_rating>0</spam_rating>
  <publish_date epoch="1341886037000">2012-07-10 14:07:17 +1200</publish_date>
  <harvest_date epoch="1341886905000">2012-07-10 14:21:45 +1200</harvest_date>
  <PostDynamicsIteration>
    <PostDynamicsDefinition>
      <fieldId>21</fieldId>
      <label>Sentiment</label>
      <shortLabel>S</shortLabel>
      <sortOrder>1</sortOrder>
      <value>
        <![CDATA[2860,0]]>
      </value>
      <exceptionValue>2860,false</exceptionValue>
      <reportValue>Neutral</reportValue>
      <tooltip/>
    </PostDynamicsDefinition>
    <reportFormattedData>
      <![CDATA[<span style="font-weight:bold; color: #FF9900; font-size: 11pt">
Sentiment: </span>Neutral  ]]]>
    </reportFormattedData>
  </PostDynamicsIteration>
</article>
</radian6_RiverOfNews_export>

```

[Resources for Data Service](#)

The following list shows other operations you can use with the Data Service.

Resources for Data Service

The following list shows other operations you can use with the Data Service.

Operations	Example
Get data by range	<pre>GET /data/topicdata/realtime/{daterangeStart} /{daterangeEnd}/{topics}/{mediatypes}/{pageIndex}/{pageSize}</pre>
Get tag cloud data	<pre>GET /data/tagclouddata/{recentXhours}/{topics} /{mediatypes}/{advancedQueryFilters}</pre>
Get tag cloud data by range	<pre>GET /data/tagclouddata/{daterangeStart} /{daterangeEnd}/{topics}/{mediatypes}/{advancedQueryFilters}</pre>
Get topic match data	<pre>GET /data/comparisondata/{recentXhours}/{topics} /{mediatypes}/{segmentation}/{countBy}</pre>
Get topic match data by range	<pre>GET /data/comparisondata/{daterangeStart} /{daterangeEnd}/{topics}/{mediatypes}/{segmentation}/{countBy}</pre>
Get widget data	<pre>GET /data/widget/{widgetId}</pre>

The `auth_token` and `auth_appkey` header parameters are required for all calls.

Blog Service

The Blog Service enables you to perform operations such as fetching a list of posts for a given site and adding notes to a site.

Get Blog Details

Fetch a list of posts for a given site including workflow details.

```
GET /blog/workflow/{blogId}/{topicId}
```

Parameters	Type	Description
<code>auth_token</code>	HeaderParam	Required. Request header containing the token returned from authentication with the API.
<code>auth_appkey</code>	HeaderParam	Required. Application key unique to your account.
<code>blogId</code>	PathParam	Blog Ids for which to get posts for.
<code>topicId</code>	PathParam	Comma delimited list of topic Ids.
<code>maxPostCount</code>	QueryParam	Total number of post to return per request.
<code>pageNum</code>	QueryParam	The page index to return posts for. For example if the total count of posts for the request exceeds the maximum number of posts (as indicated by

Parameters	Type	Description
		the maxPostCount paramter), the page number can be incremented to fetch those posts not returned in the current request.

Example

```
http://api.radian6.com/socialcloud/v1/blog/workflow/5...5/2...7
```

Request Headers

```
GET /socialcloud/v1/blog/workflow/{blogId}/{topicId} HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<radian6_RiverOfNews_export>
  <report_date>2012-07-11 02:24:04 +1200</report_date>
  <user_name>Jane.Smith@...</user_name>
  <RoN_sort_order>publishedDate</RoN_sort_order>
  <article_count>1</article_count>
  <total_article_count>1</total_article_count>
  <article ID="2...3">
    <description charset="UTF-8">
      <headline>
        <![CDATA[Post from Facebook user]]>
      </headline>
      <author fbid="17...9" externalId="17...9">
        <![CDATA[Facebook user]]>
      </author>
      <recipient>
        <![CDATA[None]]>
      </recipient>
      <content>
        <![CDATA[Facebook post content...]]>
      </content>
      <external_id>17...9_22...4</external_id>
    </description>
```



```

<source ID="1...2">
  <![CDATA[Post from Facebook user]]>
</source>
<host>
  <![CDATA[www.facebook.com]]>
</host>
<article_url>
  <![CDATA[http://www.facebook.com/permalink.php?story_fbid=22...4&id=17...9]]>

</article_url>
<media_provider>facebook.com Discussions</media_provider>
<media_type_id>12</media_type_id>
<language_id>1</language_id>
<spam_rating>0</spam_rating>
<publish_date epoch="1341886037000">2012-07-10 14:07:17 +1200</publish_date>
<harvest_date epoch="1341886905000">2012-07-10 14:21:45 +1200</harvest_date>
<PostDynamicsIteration>
  <PostDynamicsDefinition>
    <fieldId>21</fieldId>
    <label>Sentiment</label>
    <shortLabel>S</shortLabel>
    <sortOrder>1</sortOrder>
    <value>
      <![CDATA[2860,0]]>
    </value>
    <exceptionValue>2860,false</exceptionValue>
    <reportValue>Neutral</reportValue>
    <tooltip/>
  </PostDynamicsDefinition>
  <reportFormattedData>
    <![CDATA[<span style="font-weight:bold; color: #FF9900; font-size: 11pt">
Sentiment: </span>Neutral  ]]]>
  </reportFormattedData>
</PostDynamicsIteration>
</article>
</radian6_RiverOfNews_export>

```

[Resources for Blog Service](#)

The following list shows other operations you can use with the Blog Service.

Resources for Blog Service

The following list shows other operations you can use with the Blog Service.

Operations	Example
Add note to a site	POST /blog/workflow/note/{blogIdList}
Add note by post Id	POST /blog/workflow/noteByPostId/{blogPostId}
Add tag by post Id	POST /blog/workflow/tagsByPostId/{blogPostId}
Get site metrics	GET /blog/metrics/{siteId}/{topicId}

Operations	Example
Get tagged blogs	GET /blog/sourcetagged/{tags}
Remove tags and notes	POST /blog/workflow/removeTagsAndNotes/{tagAndNoteIds}

The `auth_token` and `auth_appkey` header parameters are required for all calls.

Authentication Service

The Authentication Service enables you to authenticate a user in the Radian6 system.

Responds with authentication token to be used for subsequent requests as request header called `auth_token`. `auth_appkey` must also be provided for every request.

```
GET /socialcloud/v1/auth/authenticate
```

Parameters	Type	Description
<code>auth_user</code>	HeaderParam	Required. Request header containing username.
<code>auth_pass</code>	HeaderParam	Required. Request header containing the plain text password.
<code>auth_token</code>	HeaderParam	Required. Request header containing the token returned from authentication with the API.
<code>auth_appkey</code>	HeaderParam	Required. Application key unique to your account.
<code>fields</code>	QueryParam	Comma delimited list of elements to return such as <code>userdetails</code> and <code>clientattributes</code> .

Example

```
https://api.radian6.com/socialcloud/v1/auth/authenticate
?fields=userdetails,clientattributes
```

Request Headers

```
GET /socialcloud/v1/auth/authenticate HTTP/1.1
Host: api.radian6.com
auth_user: mikemullen
auth_pass: NotARealPassword
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
```

```
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<auth>
  <token>b65e06d1b5383...</token>
  <UserDetails>
    <user>
      <userId>12345</userId>
      <clientId>99</clientId>
      <displayName><![CDATA[Mike Mullen]]></displayName>
      <emailAddress>Mike.Mullen@...</emailAddress>
      <timezone>GMT</timezone>
      <packages></packages>
      <userRoleId>1</userRoleId>
      <createdDate>Jun 22, 2010 05:18 PM</createdDate>
      <enabled>true</enabled>
      <aihUsers><aihUser>
        <userKey>84ba97...</userKey>
        <registerDate>2010</registerDate>
        <type>1</type>
      </aihUser></aihUsers>
    </user>
    <avatar userId="12345"><![CDATA[http://path-to-avatar-image.jpg]]></avatar>
    <Packages></Packages>
    <ClientAttributes>
      <attribute>
        <id>12</id>
        <description>IDLE_TIMEOUT</description>
        <value>10800000</value>
      </attribute>
      ...
    </ClientAttributes>
  </UserDetails>
</auth>
```

Lookup Service

The Lookup Service enables you to perform operations such as fetching a list of media types, languages, users, and workflow items.

Get Media Types

Fetch a list of valid media types. Media types are used to indicate the type and source of social media posts within the Radian6 API. They can be used to create source filters within the Analysis Dashboard (Topic Profile Configuration) and can also be used to filter results in the calls of the Data Service. This call shows you the name and id of all the media types in the system.

```
GET /lookup/mediaproviders
```

Parameters	Type	Description
auth_token	HeaderParam	Required. Request header containing the token returned from authentication with the API.
auth_appkey	HeaderParam	Required. Application key unique to your account.

Example

```
http://api.radian6.com/socialcloud/v1/lookup/mediaproviders
```

Request Headers

```
GET /socialcloud/v1/lookup/mediaproviders HTTP/1.1
Host: api.radian6.com
auth_token: NotARealToken
auth_appkey: NotARealAppKey
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 29 Sep 2011 17:17:16 GMT
Content-Type: application/xml
Content-Length: 705
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<MediaTypeList>
  <MediaTypeItem>
    <mediaTypeId>1</mediaTypeId>
    <mediaTypeName>Blogs</mediaTypeName>
    <displayOrder>1</displayOrder>
  </MediaTypeItem>
  <MediaTypeItem>
    <mediaTypeId>2</mediaTypeId>
    <mediaTypeName>Videos</mediaTypeName>
    <displayOrder>2</displayOrder>
  </MediaTypeItem>
  <MediaTypeItem>
    <mediaTypeId>4</mediaTypeId>
    <mediaTypeName>Images</mediaTypeName>
    <displayOrder>3</displayOrder>
  </MediaTypeItem>
  <MediaTypeItem>
    <mediaTypeId>5</mediaTypeId>
    <mediaTypeName>Mainstream News</mediaTypeName>
    <displayOrder>4</displayOrder>
  </MediaTypeItem>
</MediaTypeList>
```

```

</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>8</mediaTypeId>
  <mediaTypeName>MicroMedia</mediaTypeName>
  <displayOrder>5</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>10</mediaTypeId>
  <mediaTypeName>Forums</mediaTypeName>
  <displayOrder>6</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>9</mediaTypeId>
  <mediaTypeName>Forum Replies</mediaTypeName>
  <displayOrder>7</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>11</mediaTypeId>
  <mediaTypeName>Comments</mediaTypeName>
  <displayOrder>8</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>12</mediaTypeId>
  <mediaTypeName>Facebook</mediaTypeName>
  <displayOrder>9</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>13</mediaTypeId>
  <mediaTypeName>Aggregator</mediaTypeName>
  <displayOrder>10</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>14</mediaTypeId>
  <mediaTypeName>Buy/Sell</mediaTypeName>
  <displayOrder>11</displayOrder>
</MediaTypeItem>
<MediaTypeItem>
  <mediaTypeId>16</mediaTypeId>
  <mediaTypeName>MySpace</mediaTypeName>
  <displayOrder>13</displayOrder>
</MediaTypeItem>
</MediaTypeList>

```

[Resources for Lookup Service](#)

The following list shows other operations you can use with the Lookup Service.

Resources for Lookup Service

The following list shows other operations you can use with the Lookup Service.

Operations	Example
Get sort types	GET /lookup/sorttypes

Operations	Example
Get languages	GET /lookup/languages
Get timezones	GET /lookup/timezones
Get filter types	GET /lookup/filtertypes
Get count types	GET /lookup/counttypes
Get users	GET /lookup/users
Get tags	GET /lookup/tags
Get advanced filter types	GET /lookup/advancedfiltertypes
Get regions	GET /lookup/regions
Get projects	GET /lookup/projects
Get influencer metrics	GET /lookup/influencermetrics
Get external account types	GET /lookup/externalaccounttypes
Get extended media types	GET /lookup/extendedmediatypes
Get media group types	GET /lookup/mediagroupproviders
Get workflow	GET /lookup/workflow

The `auth_token` and `auth_appkey` header parameters are required for all calls.

Resources

Use the following resources to get more information about the Radian6 API.

- Get started with Radian6: [Social Media Marketing](#)
- Radian6 case studies: [Marketing Cloud Resources](#)