
TwitterAPI Documentation

Release

dtluna, Bricky

Aug 05, 2017

Contents

1	Accounts	3
2	Blocking	5
3	Configuration	7
4	Data structures	9
4.1	User object	9
4.2	Status object	11
4.3	Group object	13
4.4	Relationship object	13
4.5	Direct message object	14
4.6	ActivityStreams status object	16
5	Direct Messages	19
6	Favorites	21
7	Friendships	23
8	Groups	25
9	Media uploads	29
10	OAuth	31
11	Search	33
12	Statuses	35
13	Timelines	37
14	Timelines (Activity Streams)	41
15	Users	45
	TwitterAPI routing table	47

Documentation source code

GET `/api/account/verify_credentials.json`

Test if supplied user credentials are valid.

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST `/api/account/verify_credentials.json`

Same as *GET /api/account/verify_credentials.json*.

POST `/api/account/register.json`

Register a new account.

Form Parameters

- **nickname** – name of the new user
- **password** – desired password
- **confirm** – password confirmation
- **email** – (optional) email associated with the new user
- **fullname** – full name associated with the profile
- **homepage** – URL associated with the profile. Will be prepended with “http://” if not present
- **location** – the city or country describing where the user is located. The contents are not normalized or geocoded in any way
- **bio** – user’s description

Example response: *User object*

POST `/api/account/update_profile.json`

Update profile.

Form Parameters

- **name** – full name associated with the profile
- **url** – URL associated with the profile. Will be prepended with “http://” if not present
- **location** – the city or country describing where the user is located. The contents are not normalized or geocoded in any way
- **description** – user’s description
- **profile_link_color** – (optional) color of the profile links

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST /api/account/update_profile_background_image.json

Update the authenticating user’s profile background image. This method can also be used to enable or disable the profile background image.

Form Parameters

- **image** – image to upload, encoded in *multipart/form-data*

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST /api/account/update_profile_image.json

Update the authenticating user’s profile image.

Form Parameters

- **image** – image to upload, encoded in *multipart/form-data*

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST /api/blocks/create.json

Blocks the specified user from following the authenticating user. In addition the blocked user will not show in the authenticating user's mentions or timeline (unless repeated by another user). If a follow or friend relationship exists it is destroyed.

Form Parameters

- **user_id** – ID of the user for whom to return results for
- **screen_name** – handle of the user for whom to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST /api/blocks/destroy.json

Un-blocks the specified user from following the authenticating user. If relationships had existed before the block was instated, they will not be restored.

Form Parameters

- **user_id** – ID of the user for whom to return results for
- **screen_name** – handle of the user for whom to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

GET /api/statusnet/config.json

Get server configuration.

Example response:

```
{
  "attachments": {
    "file_quota": 209715200,
    "uploads": true
  },
  "group": {
    "desclimit": null
  },
  "integration": {
    "source": "social.heldscal.la"
  },
  "license": {
    "image": "/theme/licenses/cc_by_3.0_80x15.png",
    "owner": null,
    "title": "Creative Commons Attribution 3.0",
    "type": "cc",
    "url": "https://creativecommons.org/licenses/by/3.0/"
  },
  "nickname": {
    "featured": []
  },
  "notice": {
    "contentlimit": null
  },
  "profile": {
    "biolimit": null
  },
  "site": {
    "broughtby": "",
    "broughtbyurl": ""
  }
}
```

```
"closed": "0",
"email": "social@rogerbraun.net",
"fancy": true,
"inviteonly": "1",
"language": "en",
"logo": "",
"name": "social.heldscal.la",
"path": false,
"private": "0",
"server": "social.heldscal.la",
"ssl": "always",
"sslserver": null,
"textlimit": "5000",
"theme": "neo-gnu",
"timezone": "UTC"
},
"throttle": {
  "count": 20,
  "enabled": false,
  "timespan": 600
},
"url": {
  "maxnoticelength": -1,
  "maxurllength": 100
},
"xmpp": {
  "enabled": false,
  "port": false,
  "server": false,
  "user": false
}
}
```

4.1 User object

```
{
  "id": 24974,
  "name": "dtluna",
  "screen_name": "dtluna",
  "location": "Minsk, Belarus",
  "description": "Attention whore, anarcho-capitalist, degenerate",
  "profile_image_url": "https://social.heldscal.la/avatar/24974-48-20170628000001.
↪jpeg",
  "profile_image_url_https": "https://social.heldscal.la/avatar/24974-48-
↪20170628000001.jpeg",
  "profile_image_url_profile_size": "https://social.heldscal.la/avatar/24974-96-
↪20170628000001.jpeg",
  "profile_image_url_original": "https://social.heldscal.la/avatar/-original-
↪tmp20170628000001.jpeg",
  "groups_count": 23,
  "linkcolor": "974b8b",
  "backgroundcolor": "974b8b",
  "url": "https://commandandconquer.club/pages/about.html",
  "protected": false,
  "followers_count": 369,
  "friends_count": 172,
  "created_at": "Sun Sep 18 18:48:59 +0000 2016",
  "utc_offset": "10800",
  "time_zone": "Europe/Minsk",
  "statuses_count": 18167,
  "following": false,
  "statusnet_blocking": false,
  "notifications": false,
  "status": {
    "text": "!stockings https://social.heldscal.la/attachment/523001",
    "truncated": false,
```

```

    "created_at": "Wed Aug 02 09:00:03 +0000 2017",
    "in_reply_to_status_id": null,
    "uri": "tag:social.heldscal.la,2017-08-02:noticeId=3253478:objectType=note",
    "source": "GS media bot",
    "source_link": null,
    "id": 3253478,
    "in_reply_to_user_id": null,
    "in_reply_to_screen_name": null,
    "geo": null,
    "attachments": [
      {
        "url": "https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg",
        "mimetype": "image/jpeg",
        "size": "162532",
        "oembed": false,
        "id": "523001",
        "version": "1.2",
        "thumb_url": "https://social.heldscal.la/file/thumb/thumb-523001-
↪450x534-d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg",
        "large_thumb_url": "https://social.heldscal.la/file/thumb/thumb-
↪523001-809x960-d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg
↪",
        "width": "809",
        "height": "960"
      }
    ],
    "statusnet_html": "!<a href=\"https://gs.smuglo.li/group/78/id\" class=\"h-
↪card u-url p-nickname group\" title=\"Stockings and knee socks (stockings)\">
↪stockings</a> <a href=\"https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg\" title=\"
↪https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg\" rel=\"
↪nofollow external norereferrer\" class=\"attachment thumbnail\" id=\"attachment-
↪523001\">https://social.heldscal.la/attachment/523001</a>",
    "statusnet_conversation_id": 1843056,
    "statusnet_in_groups": [
      {
        "nickname": "stockings",
        "url": "https://gs.smuglo.li/group/78/id"
      }
    ],
    "external_url": "https://social.heldscal.la/notice/3253478",
    "in_reply_to_profileurl": null,
    "in_reply_to_ostatus_uri": null,
    "attentions": [],
    "fave_num": 0,
    "repeat_num": 0,
    "is_post_verb": true,
    "is_local": true,
    "object_type": "http://activitystrea.ms/schema/1.0/note",
    "favorited": false,
    "repeated": false
  },
  "statusnet_profile_url": "https://social.heldscal.la/dtluna",
  "cover_photo": "https://social.heldscal.la/avatar/24974-original-banner-
↪20160918185212.jpeg",
  "background_image": "https://social.heldscal.la/avatar/24974-original-bg-
↪20160918185248.jpeg",

```

```

    "profile_link_color": "974b8b",
    "profile_background_color": "974b8b",
    "profile_banner_url": "https://social.heldscal.la/avatar/24974-original-banner-
↪20160918185212.jpeg",
    "is_local": true,
    "is_silenced": false,
    "rights": {
      "delete_user": false,
      "delete_others_notice": false,
      "silence": false,
      "sandbox": false
    },
    "is_sandboxed": false,
    "ostatus_uri": "https://social.heldscal.la/user/24974",
    "favourites_count": 16750
  }

```

4.2 Status object

```

{
  "text": "!stockings https://social.heldscal.la/attachment/523001",
  "truncated": false,
  "created_at": "Wed Aug 02 09:00:03 +0000 2017",
  "in_reply_to_status_id": null,
  "uri": "tag:social.heldscal.la,2017-08-02:noticeId=3253478:objectType=note",
  "source": "GS media bot",
  "source_link": null,
  "id": 3253478,
  "in_reply_to_user_id": null,
  "in_reply_to_screen_name": null,
  "geo": null,
  "attachments": [
    {
      "url": "https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg",
      "mimetype": "image/jpeg",
      "size": "162532",
      "oembed": false,
      "id": "523001",
      "version": "1.2",
      "thumb_url": "https://social.heldscal.la/file/thumb/thumb-523001-450x534-
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg",
      "large_thumb_url": "https://social.heldscal.la/file/thumb/thumb-523001-
↪809x960-d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg",
      "width": "809",
      "height": "960"
    }
  ],
  "user": {
    "id": 24974,
    "name": "My Life Is A Fuck",
    "screen_name": "dtluna",
    "location": "Minsk, Belarus",
    "description": "Attention whore, anarcho-capitalist, degenerate / \nHate_
↪coservative, leftist and other puritan niggers",

```

```

    "profile_image_url": "https://social.heldscal.la/avatar/24974-48-
↪20170628000001.jpeg",
    "profile_image_url_https": "https://social.heldscal.la/avatar/24974-48-
↪20170628000001.jpeg",
    "profile_image_url_profile_size": "https://social.heldscal.la/avatar/24974-96-
↪20170628000001.jpeg",
    "profile_image_url_original": "https://social.heldscal.la/avatar/-original-
↪tmp20170628000001.jpeg",
    "groups_count": 23,
    "linkcolor": "974b8b",
    "backgroundcolor": "974b8b",
    "url": "https://commandandconquer.club/pages/about.html",
    "protected": false,
    "followers_count": 369,
    "friends_count": 172,
    "created_at": "Sun Sep 18 18:48:59 +0000 2016",
    "utc_offset": "10800",
    "time_zone": "Europe/Minsk",
    "statuses_count": 18167,
    "following": false,
    "statusnet_blocking": false,
    "notifications": false,
    "statusnet_profile_url": "https://social.heldscal.la/dtluna",
    "cover_photo": "https://social.heldscal.la/avatar/24974-original-banner-
↪20160918185212.jpeg",
    "background_image": "https://social.heldscal.la/avatar/24974-original-bg-
↪20160918185248.jpeg",
    "profile_link_color": "974b8b",
    "profile_background_color": "974b8b",
    "profile_banner_url": "https://social.heldscal.la/avatar/24974-original-
↪banner-20160918185212.jpeg",
    "is_local": true,
    "is_silenced": false,
    "rights": {
        "delete_user": false,
        "delete_others_notice": false,
        "silence": false,
        "sandbox": false
    },
    "is_sandboxed": false,
    "ostatus_uri": "https://social.heldscal.la/user/24974",
    "favourites_count": 16750
},
    "statusnet_html": "!<a href=\"https://gs.smuglo.li/group/78/id\" class=\"h-card u-
↪url p-nickname group\" title=\"Stockings and knee socks (stockings)\">stockings</a>
↪<a href=\"https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg\" title=\
↪\"https://social.heldscal.la/file/
↪d12bb30e57b402b8afdb0e6bbc8c8600b80f7227aedfb73655455c54163205ab.jpg\" rel=\
↪\"nofollow external noferrer\" class=\"attachment thumbnail\" id=\"attachment-
↪523001\">https://social.heldscal.la/attachment/523001</a>",
    "statusnet_conversation_id": 1843056,
    "statusnet_in_groups": [
        {
            "nickname": "stockings",
            "url": "https://gs.smuglo.li/group/78/id"
        }
    ]
},
],

```



```

"external_url": "https://social.heldscal.la/notice/3253478",
"in_reply_to_profileurl": null,
"in_reply_to_ostatus_uri": null,
"attentions": [],
"fave_num": 0,
"repeat_num": 0,
"is_post_verb": true,
"is_local": true,
"object_type": "http://activitystrea.ms/schema/1.0/note",
"favorited": false,
"repeated": false
}

```

4.3 Group object

```

{
  "id": 78,
  "url": "https://gs.smuglo.li/group/78/id",
  "nickname": "stockings",
  "fullname": "Stockings and knee socks",
  "admin_count": 0,
  "member_count": 40,
  "original_logo": "https://gs.smuglo.li/avatar/740-300-20160229121056.jpeg",
  "homepage_logo": "https://gs.smuglo.li/avatar/740-96-20160229121056.jpeg",
  "stream_logo": "https://gs.smuglo.li/avatar/740-48-20160229121056.jpeg",
  "mini_logo": "https://gs.smuglo.li/avatar/740-24-20160229121056.jpeg",
  "homepage": "",
  "description": "Sexy and lewd images of girls in stockings and knee_
↪socks\r\nPlease don't post pictures where I can see titties and/or pussy, cause I_
↪just want the light erotic pictures here.",
  "location": "",
  "created": "Mon Feb 29 12:08:07 +0000 2016",
  "modified": "Thu Apr 14 11:22:04 +0000 2016"
}

```

4.4 Relationship object

```

{
  "relationship": {
    "source": {
      "screen_name": "dtluna",
      "followed_by": true,
      "following": true,
      "notifications_enabled": true,
      "blocking": false,
      "id": 24974
    },
    "target": {
      "screen_name": "lambadalambda",
      "followed_by": true,
      "following": true,
      "notifications_enabled": true,
    }
  }
}

```

```
        "blocking": false,  
        "id": 23211  
    }  
}
```

4.5 Direct message object

```
{  
  "created_at": "Wed Aug 02 10:58:14 +0000 2017",  
  "id": 4,  
  "recipient": {  
    "background_image": "https://social.heldscal.la/avatar/23211-original-bg-  
↪20160727143039.jpeg",  
    "backgroundcolor": "ebb7e1",  
    "blocks_you": false,  
    "cover_photo": "https://social.heldscal.la/avatar/23211-original-banner-  
↪20160727175432.gif",  
    "created_at": "Wed Jul 27 09:50:29 +0000 2016",  
    "description": "Call me Deacon Blues.",  
    "favourites_count": 20876,  
    "followers_count": 885,  
    "following": true,  
    "follows_you": true,  
    "friends_count": 1209,  
    "groups_count": 16,  
    "id": 23211,  
    "is_local": true,  
    "is_sandboxed": false,  
    "is_silenced": false,  
    "linkcolor": "f58d2c",  
    "location": "Berlin",  
    "name": "Constance Variable",  
    "notifications": true,  
    "ostatus_uri": "https://social.heldscal.la/user/23211",  
    "profile_background_color": "ebb7e1",  
    "profile_banner_url": "https://social.heldscal.la/avatar/23211-original-  
↪banner-20160727175432.gif",  
    "profile_image_url": "https://social.heldscal.la/avatar/23211-48-  
↪20170416114255.jpeg",  
    "profile_image_url_https": "https://social.heldscal.la/avatar/23211-48-  
↪20170416114255.jpeg",  
    "profile_image_url_original": "https://social.heldscal.la/avatar/23211-  
↪original-20170416114255.jpeg",  
    "profile_image_url_profile_size": "https://social.heldscal.la/avatar/23211-96-  
↪20170416114255.jpeg",  
    "profile_link_color": "f58d2c",  
    "protected": false,  
    "rights": {  
      "delete_others_notice": false,  
      "delete_user": false,  
      "sandbox": false,  
      "silence": false  
    },  
    "screen_name": "lambdalaambda",  
  },  
}
```

```

    "statuses_count": 15639,
    "statusnet_blocking": false,
    "statusnet_profile_url": "https://social.heldscal.la/lambadalambda",
    "time_zone": "UTC",
    "url": null,
    "utc_offset": "0"
  },
  "recipient_id": 23211,
  "recipient_screen_name": "lambadalambda",
  "sender": {
    "background_image": "https://social.heldscal.la/avatar/24974-original-bg-
↪20160918185248.jpeg",
    "backgroundcolor": "974b8b",
    "blocks_you": false,
    "cover_photo": "https://social.heldscal.la/avatar/24974-original-banner-
↪20160918185212.jpeg",
    "created_at": "Sun Sep 18 18:48:59 +0000 2016",
    "description": "Attention whore, anarcho-capitalist, degenerate / \nHate_
↪coservative, leftist and other puritan niggers",
    "favourites_count": 16750,
    "followers_count": 369,
    "following": true,
    "follows_you": true,
    "friends_count": 172,
    "groups_count": 23,
    "id": 24974,
    "is_local": true,
    "is_sandboxed": false,
    "is_silenced": false,
    "linkcolor": "974b8b",
    "location": "Minsk, Belarus",
    "name": "My Life Is A Fuck",
    "notifications": true,
    "ostatus_uri": "https://social.heldscal.la/user/24974",
    "profile_background_color": "974b8b",
    "profile_banner_url": "https://social.heldscal.la/avatar/24974-original-
↪banner-20160918185212.jpeg",
    "profile_image_url": "https://social.heldscal.la/avatar/24974-48-
↪20170628000001.jpeg",
    "profile_image_url_https": "https://social.heldscal.la/avatar/24974-48-
↪20170628000001.jpeg",
    "profile_image_url_original": "https://social.heldscal.la/avatar/-original-
↪tmp20170628000001.jpeg",
    "profile_image_url_profile_size": "https://social.heldscal.la/avatar/24974-96-
↪20170628000001.jpeg",
    "profile_link_color": "974b8b",
    "protected": false,
    "rights": {
      "delete_others_notice": false,
      "delete_user": false,
      "sandbox": false,
      "silence": false
    }
  },
  "screen_name": "dtluna",
  "statuses_count": 18167,
  "statusnet_blocking": false,
  "statusnet_profile_url": "https://social.heldscal.la/dtluna",
  "time_zone": "Europe/Minsk",

```

```
    "url": "https://commandandconquer.club/pages/about.html",
    "utc_offset": "10800"
  },
  "sender_id": 24974,
  "sender_screen_name": "dtluna",
  "text": "testing and stuff"
}
```

4.6 ActivityStreams status object

```
{
  "generator": "postActiv 1.0.2-dev",
  "title": "social.heldscal.la public timeline",
  "totalItems": 1,
  "items": [
    {
      "actor": {
        "id": "https://social.heldscal.la/user/29929",
        "displayName": "biancabot",
        "status_net": {
          "avatarLinks": [
            {
              "url": "https://social.heldscal.la/avatar/29929-original-
↪20170630051132.jpeg",
              "rel": "avatar",
              "type": "image/jpeg",
              "width": 1040,
              "height": 1040
            },
            {
              "url": "https://social.heldscal.la/avatar/29929-96-
↪20170630051132.jpeg",
              "rel": "avatar",
              "type": "image/jpeg",
              "width": 96,
              "height": 96
            },
            {
              "url": "https://social.heldscal.la/avatar/29929-48-
↪20170630051132.jpeg",
              "rel": "avatar",
              "type": "image/jpeg",
              "width": 48,
              "height": 48
            },
            {
              "url": "https://social.heldscal.la/avatar/29929-24-
↪20170630051133.jpeg",
              "rel": "avatar",
              "type": "image/jpeg",
              "width": 24,
              "height": 24
            }
          ]
        }
      },
      "profile_info": {
```

```

        "local_id": "29929",
        "following": "true",
        "blocking": "false"
    },
    },
    "image": {
        "url": "https://social.heldscal.la/avatar/29929-96-20170630051132.
↪ jpeg",
        "rel": "avatar",
        "type": "image/jpeg",
        "width": 96,
        "height": 96
    },
    "objectType": "person",
    "url": "https://social.heldscal.la/biancabot",
    "followers": {
        "url": "https://social.heldscal.la/biancabot/subscribers"
    },
    "portablecontacts_net": {
        "preferredUsername": "biancabot",
        "displayName": "biancabot"
    }
},
"content": "!<a href=\"https://sealion.club/group/541/id\" class=\"h-card
↪ u-url p-nickname group\" title=\"Lewds and Bewbs (lewds)\">lewds</a> #<span class=
↪ \"tag\"><a href=\"https://social.heldscal.la/tag/nsfw\" rel=\"tag\">nsfw</a></span> #
↪ <span class=\"tag\"><a href=\"https://social.heldscal.la/tag/bianca\" rel=\"tag\">
↪ bianca</a></span> <a href=\"https://social.heldscal.la/file/
↪ 8028c60b30ddfd39126cf6b615868d12b95133b18746aa994960f8e66741d379.jpg\" title=
↪ \"https://social.heldscal.la/file/
↪ 8028c60b30ddfd39126cf6b615868d12b95133b18746aa994960f8e66741d379.jpg\" rel=
↪ \"nofollow external norereferrer\" class=\"attachment\" id=\"attachment-664828\">
↪ https://social.heldscal.la/attachment/664828</a>\",
    "generator": {
        "id": "tag:social.heldscal.la,2017-08-02:notice-source:GS media bot",
        "objectType": "application",
        "status_net": {
            "source_code": "GS media bot"
        }
    },
    "id": "tag:social.heldscal.la,2017-08-02:noticeId=3254370:objectType=note
↪ ",
    "object": {
        "id": "tag:social.heldscal.la,2017-08-
↪ 02:noticeId=3254370:objectType=note",
        "objectType": "note",
        "content": "!<a href=\"https://sealion.club/group/541/id\" class=\"h-
↪ card u-url p-nickname group\" title=\"Lewds and Bewbs (lewds)\">lewds</a> #<span
↪ class=\"tag\"><a href=\"https://social.heldscal.la/tag/nsfw\" rel=\"tag\">nsfw</a></
↪ span> #<span class=\"tag\"><a href=\"https://social.heldscal.la/tag/bianca\" rel=
↪ \"tag\">bianca</a></span> <a href=\"https://social.heldscal.la/file/
↪ 8028c60b30ddfd39126cf6b615868d12b95133b18746aa994960f8e66741d379.jpg\" title=
↪ \"https://social.heldscal.la/file/
↪ 8028c60b30ddfd39126cf6b615868d12b95133b18746aa994960f8e66741d379.jpg\" rel=
↪ \"nofollow external norereferrer\" class=\"attachment\" id=\"attachment-664828\">
↪ https://social.heldscal.la/attachment/664828</a>\",
        "url": "https://social.heldscal.la/notice/3254370",
        "status_net": {

```

```
        "notice_id": 3254370
      },
      "tags": [
        {
          "objectType": "http://activityschema.org/object/hashtag",
          "displayName": "bianca"
        },
        {
          "objectType": "http://activityschema.org/object/hashtag",
          "displayName": "nsfw"
        }
      ]
    },
    "to": [
      {
        "objectType": "http://activitystrea.ms/schema/1.0/group",
        "id": "https://sealion.club/group/541/id"
      },
      {
        "objectType": "http://activitystrea.ms/schema/1.0/collection",
        "id": "http://activityschema.org/collection/public"
      }
    ],
    "status_net": {
      "conversation": "tag:social.heldscal.la,2017-08-
↪02:objectType=thread:nonce=d36c95fe72587732",
      "notice_info": {
        "local_id": "3254370",
        "source": "GS media bot",
        "repeated": "false",
        "favorite": "false"
      }
    },
    "published": "2017-08-02T11:00:04+00:00",
    "provider": {
      "objectType": "service",
      "displayName": "social.heldscal.la",
      "url": "https://social.heldscal.la/"
    },
    "verb": "post",
    "url": "https://social.heldscal.la/notice/3254370"
  }
],
"links": [
  {
    "url": "https://social.heldscal.la/main/public",
    "rel": "alternate",
    "type": "text/html"
  }
]
}
```

Direct Messages

GET `/api/direct_messages.json`

Returns the 20 most recent direct messages sent to the authenticating user. You can request up to 200 direct messages per call, and only the most recent 200 DMs will be available using this endpoint.

Query Parameters

- **since_id** (*int*) – returns direct messages with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns direct messages with an ID less than (that is, older than) or equal to the specified ID
- **count** (*int*) – number of direct messages to try and retrieve

Example response: *Direct message object*

GET `/api/direct_messages/sent.json`

Returns the 20 most recent direct messages sent by the authenticating user. You can request up to 200 direct messages per call, and only the most recent 200 DMs will be available using this endpoint.

Query Parameters

- **since_id** (*int*) – returns direct messages with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns direct messages with an ID less than (that is, older than) or equal to the specified ID
- **count** (*int*) – number of direct messages to try and retrieve

Example response: *Direct message object*

POST `/api/direct_messages/new.json`

Sends a new direct message to the specified user from the authenticating user.

Form Parameters

- **text** – text of your message
- **user_id** – the ID of the recipient

- **screen_name** – the handle of the recipient

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Direct message object*

GET `/api/favorites.json`

Returns recent notices favorited by the authenticating or specified user.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for
- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *status objects*

POST `/api/favorites/create/(int: status_id).json`

Favorites a status.

Parameters

- **status_id** (*int*) – ID of the status to favorite

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Status object*

POST `/api/favorites/destroy/(int: status_id).json`

Unfavorites a status.

Parameters

- **status_id** (*int*) – ID of the status to unfavorite

Request Headers

- *Authorization* – username and password or OAuth token to authenticate

Example response: *Status object*

POST /api/friendships/create.json

Subscribe to status updates from specified user.

Form Parameters

- **user_id** – ID of the user for whom to return results for
- **screen_name** – handle of the user for whom to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

POST /api/friendships/destroy.json

Unsubscribe to status updates from specified user.

Form Parameters

- **user_id** – ID of the user for whom to return results for
- **screen_name** – handle of the user for whom to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *User object*

GET /api/friendships/exists.json

Show if `source_user` follows `target_user`.

Query Parameters

- **source_user** – user that is following. Can be an ID or username
- **target_user** – user that is following. Can be an ID or username

Example response:

```
true
```

GET `/api/friendships/show.json`

Show detailed information about the relationship between two users.

Query Parameters

- **source_id** (*int*) – (optional) ID of the subject user
- **source_screen_name** (*str*) – (optional) handle of the subject user
- **target_id** (*int*) – (optional) ID of the target user
- **target_screen_name** (*str*) – (optional) handle of the target user

Example response: *Relationship object*

GET `/api/statusnet/groups/timeline/` (**int:** *id*)
string: `nickname.json` Show a group's timeline. Similar to other timeline resources.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*string*) – handle of the group for which to return results for

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *status objects*

GET `/api/statusnet/groups/show/` (**int:** *id*)
string: `nickname.json` Show details about the group.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*string*) – handle of the group for which to return results for

Example response: *Group object*

POST `/api/statusnet/groups/join/` (**int:** *id*)
string: `nickname.json` Join a group.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*string*) – handle of the group for which to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Group object*

POST `/api/statusnet/groups/leave/ (int: id)`
string: `nickname.json` Leave a group.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*string*) – handle of the group for which to return results for

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Group object*

POST `/api/statusnet/groups/create.json`
Create a group.

Form Parameters

- **nickname** – name of the new group
- **full_name** – (optional) full name associated with the group
- **homepage** – (optional) home page URL associated with the group
- **location** – (optional) The city or country describing where the group is located. The contents are not normalized or geocoded in any way
- **description** – (optional) A description of the group
- **aliases** – (optional) aliases that group has

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Group object*

GET `/api/statusnet/groups/list_all.json`
List all local groups.

Query Parameters

- **count** (*int*) – number of groups to try and retrieve

Example response: a list of *group objects*

GET `/api/statusnet/groups/list.json`
Show the groups which the specified user is a member of.

Query Parameters

- **count** (*int*) – number of groups to try and retrieve
- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*str*) – handle of the user for whom to return results for

Example response: a list of *group objects*

GET `/api/statusnet/groups/membership/ (int: id)`
string: `nickname.json` List the members of the specified group.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*str*) – handle of the group for which to return results for

Query Parameters

- **count** (*int*) – number of users to try and retrieve

Example response: a list of *user objects*

GET /api/statusnet/groups/is_member.json

Show if the specified user is a member of the specified group.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*str*) – handle of the user for whom to return results for
- **group_id** (*int*) – (optional) ID of the group for which to return results for
- **group_name** (*str*) – (optional) handle of the group for which to return results for

Example response:

```
{"is_member": true}
```

GET /api/statusnet/groups/admins/(int: id)

string: *nickname.json* List the admins of the specified group.

Parameters

- **id** (*int*) – ID of the group for which to return results for
- **nickname** (*str*) – handle of the group for which to return results for

Query Parameters

- **count** (*int*) – number of users to try and retrieve

Example response: a list of *user objects*

Media uploads

POST /api/statusnet/media/upload

Upload a file.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok" xmlns:atom="http://www.w3.org/2005/Atom">
  <mediaid>573169</mediaid>
  <mediaurl>https://social.heldscal.la/attachment/573169</mediaurl>
  <media_url>https://social.heldscal.la/attachment/573169</media_url>
  <size>35586</size>
  <atom:link rel="enclosure" href="https://social.heldscal.la/file/
  ↪8f4757d1cf693eblace0fb904b1fcaab1805cac32bda1192de1ae7b8f199e235.png" type=
  ↪"image/png"></atom:link>
  <media_id>573169</media_id>
  <media_id_string>573169</media_id_string>
  <image w="514" h="403" image_type="image/png"></image>
</rsp>
```

Form Parameters

- **media** – file to upload, encoded in *multipart/form-data*

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

POST /api/media/upload.json

Upload a file.

Form Parameters

- **media** – file to upload, encoded in *multipart/form-data*

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response:

```
{
  "media_id": "664893",
  "media_id_string": "664893",
  "media_url": "https://social.heldscal.la/attachment/664893",
  "size": "13"
}
```

POST /api/oauth/request_token

Allows a Consumer application to obtain an OAuth Request Token to request user authorization. This method fulfills Section 6.1 of the OAuth 1.0 authentication flow. It is strongly recommended you use HTTPS for all OAuth authorization steps.

Form Parameters

- **oauth_callback** – URL for callback or oob

Request Headers

- **Authorization** – OAuth token to authenticate

Example response:

```
oauth_token=b41de9b1026dfa4506f043f3c50fea21&oauth_token_
↪secret=29f365bba6ccd55d72703f75ee1f1f&oauth_callback_confirmed=true
```

GET /api/oauth/authorize

Allows a Consumer application to use an OAuth Request Token to request user authorization. This method fulfills Section 6.2 of the OAuth 1.0 authentication flow.

Query Parameters

- **oauth_token** (*string*) – OAuth token to authenticate

In response you will get a web page to allow access for an application. After allowing it, you will get a security code.

POST /api/oauth/access_token

Allows a Consumer application to exchange the OAuth Request Token for an OAuth Access Token. This method fulfills Section 6.3 of the OAuth 1.0 authentication flow. The OAuth access token may also be used for xAuth operations.

Request Headers

- **Authorization** – OAuth token to authenticate

Example response:

```
oauth_token=fb05a0d00257gf0e1cb39f0c8d59575e&oauth_token_  
↪secret=ab78cf28ca6536ae78f4082d4e1a0331
```

Todo: research /api/oauth/authenticate

GET /api/search.json

Get a collection of statuses matching a specified query.

Query Parameters

- **query** (*str*) – UTF-8 encoded query
- **page** (*int*) – (optional) the page number (starting at 1) to return
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **rpp** (*int*) – (optional) the number of notices to return per page, up to a max of 100
- **callback** (*str*) – if supplied when using the JSON format, the response will use the JSONP format with a callback of the given name

Example response: a list of *status objects*

POST /api/statuses/update.json

Post a new status.

Form Parameters

- **status** – the text of your status update
- **source** – (optional) the name of application you update the status from
- **in_reply_to_status_id** – (optional) the ID of an existing status that the update is in reply to
- **lat** – (optional) the latitude of the location this status refers to. This parameter will be ignored unless it is inside the range -90.0 to +90.0 (North is positive) inclusive. It will also be ignored if there isn't a corresponding `long` parameter
- **long** – (optional) the longitude of the location this status refers to. The valid ranges for longitude is -180.0 to +180.0 (East is positive) inclusive. This parameter will be ignored if outside that range, if it is not a number or if there not a corresponding `lat` parameter
- **media** – (optional) image to upload, encoded in *multipart/form-data*

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Status object*

POST /api/statuses/destroy/(int: status_id).json

Delete the status specified by the required ID parameter. The authenticating user must be the author of the specified status. The destroyed status if successful is returned.

Parameters

- **status_id** (*int*) – the ID of the status to delete

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Status object*

GET `/api/statuses/show/(int: status_id).json`

Show a specified status.

Parameters

- **status_id** (*int*) – the ID of the status to show

Example response: *Status object*

POST `/api/statuses/retweet/(int: status_id).json`

Repeat a status.

Parameters

- **status_id** (*int*) – the ID of the status to repeat

Request Headers

- **Authorization** – username and password or OAuth token to authenticate

Example response: *Status object*

GET `/api/statuses/conversation/(int: conversation_id).json`

Show statuses that have been posted in the conversation.

Parameters

- **conversation_id** (*int*) – the ID of the conversation to show

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *status objects*

GET `/api/statuses/public_timeline.json`

Show the most recent notices, including repeats if they exist, from non-protected users.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *status objects*

GET `/api/statuses/home_timeline.json`

Show the most recent notices, including repeats if they exist, posted by the authenticating user and the users they follow.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *status objects*

GET `/api/statuses/friends_timeline.json`

Show the most recent notices, including repeats if they exist, posted by the authenticating or specified user and the users they follow.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve

- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *status objects*

GET /api/statuses/user_timeline.json

Show the most recent notices posted by the authenticating or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *status objects*

GET /api/statuses/mentions.json

GET /api/statuses/mentions_timeline.json

Show the most recent mentions (notices containing @username) for the authenticating user or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *status objects*

GET /api/statuses/replies.json

Show the most recent mentions (notices containing @username) for the authenticating user or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *status objects*

Timelines (Activity Streams)

GET `/api/statuses/public_timeline.as`

Show the most recent notices, including repeats if they exist, from non-protected users.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *ActivityStream status objects*

GET `/api/statuses/home_timeline.as`

Show the most recent notices, including repeats if they exist, posted by the authenticating user and the users they follow.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID

Example response: a list of *ActivityStream status objects*

GET `/api/statuses/friends_timeline.as`

Show the most recent notices, including repeats if they exist, posted by the authenticating or specified user and the users they follow.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve

- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *ActivityStream status objects*

GET /api/statuses/user_timeline.as

Show the most recent notices posted by the authenticating or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *ActivityStream status objects*

GET /api/statuses/mentions.as

GET /api/statuses/mentions_timeline.as

Show the most recent mentions (notices containing @username) for the authenticating user or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *ActivityStream status objects*

GET /api/statuses/replies.as

Show the most recent mentions (notices containing @username) for the authenticating user or specified user.

Query Parameters

- **count** (*int*) – number of statuses to try and retrieve
- **since_id** (*int*) – returns statuses with an ID greater than (that is, more recent than) the specified ID
- **max_id** (*int*) – returns statuses with an ID less than (that is, older than) or equal to the specified ID
- **user_id** (*int*) – ID of the user whose timeline to show
- **screen_name** (*string*) – handle of the user whose timeline to show

Example response: a list of *ActivityStream status objects*

GET `/api/users/show.json`

Show detailed information about the specified user.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for

Example response: *User object*

GET `/api/statuses/followers.json`

Show followers of the specified user.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for

Example response: *User object*

GET `/api/statuses/friends.json`

Show users the specified user follows.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for

Example response: *User object*

GET `/api/friends/ids.json`

Show IDs of users the specified user follows.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for

Example response:

```
[51450, 50736, 51295, 50959]
```

GET `/api/followers/ids.json`

Show IDs of followers of the specified user.

Query Parameters

- **user_id** (*int*) – ID of the user for whom to return results for
- **screen_name** (*string*) – handle of the user for whom to return results for

Example response:

```
[51450, 50736, 51295, 50959]
```

Todo: describe errors

Todo: describe errors

original entry

Todo: research `/api/oauth/authenticate`

original entry

TwitterAPI routing table

/api

GET /api/account/verify_credentials.json, 3	38
GET /api/direct_messages.json, 19	GET /api/statusnet/config.json, 7
GET /api/direct_messages/sent.json, 19	GET /api/statusnet/groups/admins/(int:id) (string:nic), 27
GET /api/favorites.json, 21	GET /api/statusnet/groups/is_member.json, 27
GET /api/followers/ids.json, 46	GET /api/statusnet/groups/list.json, 26
GET /api/friends/ids.json, 45	GET /api/statusnet/groups/list_all.json, 26
GET /api/friendships/exists.json, 23	GET /api/statusnet/groups/membership/(int:id) (string:nic), 26
GET /api/friendships/show.json, 24	GET /api/statusnet/groups/show/(int:id) (string:nic), 25
GET /api/oauth/authorize, 31	GET /api/statusnet/groups/timeline/(int:id) (string:nic), 25
GET /api/search.json, 33	GET /api/users/show.json, 45
GET /api/statuses/conversation/(int:conversation_id).json, 36	POST /api/account/register.json, 3
GET /api/statuses/followers.json, 45	POST /api/account/update_profile.json, 3
GET /api/statuses/friends.json, 45	POST /api/account/update_profile_background_image.json, 4
GET /api/statuses/friends_timeline.as, 41	POST /api/account/update_profile_image.json, 4
GET /api/statuses/friends_timeline.json, 37	POST /api/account/verify_credentials.json, 3
GET /api/statuses/home_timeline.as, 41	POST /api/blocks/create.json, 5
GET /api/statuses/home_timeline.json, 37	POST /api/blocks/destroy.json, 5
GET /api/statuses/mentions.as, 42	POST /api/direct_messages/new.json, 19
GET /api/statuses/mentions.json, 38	POST /api/favorites/create/(int:status_id).json, 21
GET /api/statuses/mentions_timeline.as, 42	POST /api/favorites/destroy/(int:status_id).json, 21
GET /api/statuses/mentions_timeline.json, 38	POST /api/friendships/create.json, 23
GET /api/statuses/public_timeline.as, 41	POST /api/friendships/destroy.json, 23
GET /api/statuses/public_timeline.json, 37	POST /api/media/upload.json, 29
GET /api/statuses/replies.as, 42	POST /api/oauth/access_token, 31
GET /api/statuses/replies.json, 38	POST /api/oauth/request_token, 31
GET /api/statuses/show/(int:status_id).json, 36	POST /api/statuses/destroy/(int:status_id).json, 35
GET /api/statuses/user_timeline.as, 42	POST /api/statuses/retweet/(int:status_id).json,
GET /api/statuses/user_timeline.json,	

36
POST /api/statuses/update.json, 35
POST /api/statusnet/groups/create.json,
26
POST /api/statusnet/groups/join/(int:id) (string:nickname).json,
25
POST /api/statusnet/groups/leave/(int:id) (string:nickname).json,
26
POST /api/statusnet/media/upload, 29