

GET /users

Description: returns a user's profile.

Parameter: user_id

Example Request: GET /api.nanotwitter.com/0.1/users.json?user_id=10

Example Response:

```
{
  {
    id: 12
    email: nanotwitter@gmail.com
    username: nt
    password: hidden11
    profile: http://regscqrxe.sfdgs.sdfasa.jpg
    regist_date: 2015-10-19 23:24:08 UTC
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-20 14:02:52 UTC
  }
}
```

POST /users/create

Description: Creates a new user.

Parameter: username, password

Example Request: POST /api.nanotwitter.com/0.1/users/create.json?
username="Liangwei"&password="I am password."

Example Response:

```
{
  {
    id: 12
    email: null
    username: liangwei
    password: "I am password"
    profile: default_image
    regist_date: 2015-10-19 23:24:08 UTC
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-20 14:02:52 UTC
  }
}
```

POST /users/update_account

Description: update a user's account.

Parameter: username(optional),
email(optional),password(optional),profile(optional)

Example Request: POST /api.nanotwitter.com/0.1/users/
update_account.json?password="I am another password"

Example Response:

```
{
  {
    id: 12
    email: nil
    username: liangwei
    password: "I am another password"
    profile: default_image
    regist_date: 2015-10-19 23:24:08 UTC
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-20 14:02:52 UTC
  }
}
```

GET /tweets/

Description: Returns all tweets of a user or most recent 50 tweets of all users if no parameter is provided.

Parameter: user_id(optional)

Example Request: GET /api.nanotwitter.com/0.1/tweets.json?user_id=3

Example Response:

```
{
  {
    id: 5
    user_id: 3
    content: "what is going on, buddy?"
    retweet_id: null
    pub_time: 2015-10-19 23:24:08 UTC
    media_url: http://regscqrxe.sfdgs.sdfasa.jpg
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-19 23:24:08 UTC
  },
  {
    id: 6
    user_id: 3
    content: "Feeling happy?"
    retweet_id: null
    pub_time: 2015-10-19 23:24:08 UTC
    media_url: http://regscqrxe.sfdgs.sdfasa.jpg
    created_at: 2015-10-19 23:24:08 UTC
  }
}
```

```
        updated_at: 2015-10-19 23:24:08 UTC
      },
      ...
    }
  }
```

POST /tweets/create

Description: create a tweet for authenticated user.

Parameter: content,url(optional),retweet_id(optional)

Example Request: POST /api.nanotwitter.com/0.1/tweets/create.json?
content="I am a tweet!"&url="www.pitos.com/image"

Example Response:

```
{
  {
    id: 8
    user_id: 3
    content: "I am a tweet"
    retweet_id: null
    pub_time: 2015-10-19 23:24:08 UTC
    media_url: www.pitos.com/image
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-19 23:24:08 UTC
  }
}
```

GET /favorites/list

Description: return all tweets(or specified number) favorited by the authenticating user or specified user.

Parameter: user_id(optional),count(optional)

Example Request: GET /api.nanotwitter.com/0.1/favorites/list.json?
user_id=6&count=1

Example Response:

```
{
  {
    id: 8
    user_id: 6
    content: "I am a tweet"
    retweet_id: null
    pub_time: 2015-10-19 23:24:08 UTC
    media_url: www.pitos.com/image
  }
}
```

```
        created_at: 2015-10-19 23:24:08 UTC
        updated_at: 2015-10-19 23:24:08 UTC
    }
}
```

POST /favorites/create

Description: create a favorite record between the authenticating user and specified tweet in parameter.

Parameters: tweet_id

Example Request: POST /api.nanotwitter.com/0.1/favorites/create.json?tweet_id=6

Example Response:

```
{
  {
    id: 12
    user_id: 5
    tweet_id: 3
  }
}
```

POST /favorites/destroy

Description: destroy a favorite record between the authenticating user and specified tweet in parameter and return this tweet.

Parameters: tweet_id

Example Request: POST /api.nanotwitter.com/0.1/favorites/destroy.json?tweet_id=6

Example Response:

```
{
  {
    id: 6
    user_id: 2
    content: "I am a tweet"
    retweet_id: null
    pub_time: 2015-10-19 23:24:08 UTC
    media_url: www.pitos.com/image
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-19 23:24:08 UTC
  }
}
```

GET /followings/list

Description: return all users followed by authenticating or specified user.

Parameters: user_id(optional)

Example Request: GET /api.nanotwitter.com/0.1/followings/list.json?user_id=6

Example Response:

```
{
  {
    id: 10
    email: null
    username: liangwei
    password: "I am password"
    profile: default_image
    regist_date: 2015-10-19 23:24:08 UTC
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-20 14:02:52 UTC
  },
  {
    id: 12
    email: null
    username: meiyue
    password: "I am another password"
    profile: default_image
    regist_date: 2015-10-19 23:24:08 UTC
    created_at: 2015-10-19 23:24:08 UTC
    updated_at: 2015-10-20 14:02:52 UTC
  }
  ...
}
```

POST /followings/create

Description: follow a certain user for authenticated user.

Parameters: user_id

Example Request: POST /api.nanotwitter.com/0.1/following/create.json?user_id=6

Example Response:

```
{
```

```
{
    id: 10
    follower_id: 5
    followee_id: 3
}
```

POST /followings/destroy

Description: unfollow a certain user for authenticated user and return the profile of this user.

Parameters: user_id

Example Request: POST /api.nanotwitter.com/0.1/following/create.json?user_id=6

Example Response:

```
{
    {
        id: 6
        email: null
        username: liangwei
        password: "I am password"
        profile: default_image
        regist_date: 2015-10-19 23:24:08 UTC
        created_at: 2015-10-19 23:24:08 UTC
        updated_at: 2015-10-20 14:02:52 UTC
    }
}
```

GET /followers/list

Description: return all users following authenticating or specified user.

Parameters: user_id(optional)

Example Request: GET /api.nanotwitter.com/0.1/followers/list.json?user_id=6

Example Response:

```
{
    {
        id: 10
        email: null
        username: liangwei
    }
}
```

```
password: "I am password"
profile: default_image
regist_date: 2015-10-19 23:24:08 UTC
created_at: 2015-10-19 23:24:08 UTC
updated_at: 2015-10-20 14:02:52 UTC
},
{
  id: 12
  email: null
  username: meiyue
  password: "I am another password"
  profile: default_image
  regist_date: 2015-10-19 23:24:08 UTC
  created_at: 2015-10-19 23:24:08 UTC
  updated_at: 2015-10-20 14:02:52 UTC
}
...
}
```