API v1

Response examples：

```
{
	code:  “success” ,
	message: {
		id: id,
		name: name,
		email: email
	}
}
{
	code:  “error” ,
	message:  “error message”
}
```

Notice:
1)The output parameters defined below are all in “message” attribute. Only successful situation is considered.
2) “/api/v1” in front of all routes

GET /users/:id

Requires authentication? Yes

Description: get user info by id

Input Parameters

Input Parameters	Description
id(required)	id of the user being searched
fields(optional)	Format: fields=[“name” ,” email” ,” gender”]

Output Parameters

Output Parameters	Description
name	User name

email	User email
birthday	User birthday
gender	User gender
create_time(optional)	The register time of the user. Type: int

POST /users/login

Requires authentication? No

Description: user login by username and password.

Input Parameters

Input Parameters	Description
name(required)	User name
password(required)	User password

Output Parameters

Output Parameters	Description
access_token	The access_token user get, Type: string

POST /users

Requires authentication? No

Description: create a new user and automatically do login.

Input Parameters

Input Parameters	Description
name(required)	User name
password(required)	User password
password2(required)	User password again for confirmation
Email(required)	User email

Output Parameters

Output Parameters	Description
access_token	The access_token user get, Type: string

PUT /users/:id

Require authentication? Yes

Description: user modify their personal information except password.

Input parameters

Input Parameters	Description
email(optional)	user email
gender(optional)	user gender
birthday(optional)	user birthday
name(optional)	user name

Output parameters

Output Parameters	Description
name	user name after modification

email	user email after modification
gender	user gender after modification
birthday	user birthday after modification

POST /follows

Required authentication? Yes

Description: user add a new one-way relationship

Input Parameters

Input Parameters	Description
following_id(required)	the id of the user be followed

Output Parameters

Output Parameters	Description
<User> new_following	the new following user of the active user

DELETE /follows

Required authentication: Yes

Description: cancel following to a user

Input Parameters

Input Parameters	Description
following_id(required)	user id to be removed from the following list

Output Parameters

Output Parameters	Description
none	none

GET /users/:id/followings

Require authentication? Yes

Description: get a list of users that user(id) is following. (user id may not be the active user)

Input Parameters

Input Parameters	Description
id(required)	the id of the user who owns the following list (may not be the active one)

Output Parameters

Output Parameters	Description
List<User> followings	a list of users that is followed by the user(id)

GET /users/:id/followers

Require authentication: Yes

Description: get a list of users who are currently following user(id). (user id may not be the active user)

Input Parameters

Input Parameters	Description
id(required)	the id of the user who owns the follower list (may not be the active one)

Output Parameters

Output Parameters	Description
List<User> followers	a list of users who are currently following user(id)

POST /tweets

Require authentication: Yes

Description: post a tweet (maybe it is a reply to another one)

Input Parameters

Input Parameters	Description
content(required)	the content of the tweet
reply_to_tweet_id	If this tweet is a comment, this parameter points to the original tweet. Otherwise, it is null.

Output Parameters

Output Parameters

Output Parameters	Description
<Tweet> tweet	the tweet just created

GET /tweets/recent

Require authentication: No

Description: get a list of tweets by some {start} and {number}. It is a pagination query and the result is sort by chronological order automatically. Can also be searched by tags

Input Parameters

Input Parameters	Description
start(required)	the number of the starting index
number(required)	the number of tweets will be returned
tag_name(optional)	tag name. if it is provided, then the result will only contains tweets with tag_name

Output Parameters

Output Parameters	Description
List<Tweet> tweets	the recent tweets in chronological order starting from {start} with length {number}

GET /users/:id/tweets

Require authentication: Yes

Description: get a list a tweets which is posted by a specific user who may not be the active user. It is a pagination query and the result is sort by chronological order.

Input Parameters

Input Parameters	Description
id(required)	the user who is the owner of the tweet list
start(required)	the starting index
number(required)	the number of the tweets to be returned

Output Parameters

Output Parameters	Description
List<Tweet> tweets	the recent tweets of user(id) in chronological order starting from {start} with length {number}

GET /tweets/:id

Require authentication: Yes

Description: return a single tweet by the tweet id

Input Parameters

Input Parameters	Description
id(required)	the tweet id

Output Parameters

Output Parameters	Description
<Tweet> tweet	the tweet(id)

GET /homeline

Require authentication: Yes

Description: get the time line on the home page (all the tweets from his/her followings in chronological order) of the active user.

Input Parameter

Input Parameters	Description
start(required)	the starting index
number(required)	the number of the tweets to be returned

Output Parameter

Output Parameters	Description
List<Tweet> tweets	all the tweets from his/her followings in chronological order

POST /likes

Require authentication: Yes

Description: the active user likes a specific tweet.

Input Parameters

Input Parameters	Description
tweet_id(required)	the tweet id that the active user likes

Output Parameters

Output Parameters	Description
likes	the number of likes the tweet gets

Delete /likes

Require authentication: Yes

Description: the active user cancels his/her like toward a specific tweet.

Input Parameters

Input Parameters	Description
tweet_id(required)	the tweet id that the active user likes

Output Parameters

Output Parameters	Description
likes	the number of likes the tweet gets

POST /notifications

Require authentication: Yes

Description: create a notification. (but I think notification is generated by system not by users. TBD)

Input Parameters:

Input Parameters	Description
type(required)	“mention” ， “reply” ， “new_follower”

tweet_id(optional)	if it is a mention or reply, tweet_id points to that tweet, if it is not, then tweet_id equals null
new_follower_id(optional)	if type is a “new_follower” , this points to the follower id, otherwise, it is null
target_user_id(required)	who this notification sends to

Output Parameters

Output Parameters	Description
<Notification> notification	the notification just created

GET /notifications/:id

Require authentication: Yes

Description: get a notification by id

Input Parameters

Input Parameters	Description
id(required)	the id of the notification

Output Parameters

Output Parameters	Description
<Notification> notification	the notification(id)

GET /notifications

Require authentication: Yes

Description: get a list of notifications of the active user by chronological order

Input Parameter

Input Parameters	Description
none	none

Output Parameter

Input Parameters	Description
List<Notifications> notifications	a list of notifications in chronological order

PUT /notifications/:id

Require authentication: Yes

Description: modify the status of a notification (this notification must belong to the active user)

Input Parameter

Input Parameters	Description
id	notification id
status	“read” ,” unread”

Output Parameter

Output Parameters	Description

<Notification> notification	the notification after being modified

DELETE /notifications/:id

Require authentication: Yes

Description: delete a notification (this notification must belong to the active user)

Input Parameter

Input Parameters	Description
id	notification id

Output Parameter

Output Parameters	Description
none	none

POST /tags

Require authentication: Yes

Description: create a new tag in a tweet (I think this will not be invoked by the user, but will only happens behind the scene. TBD)

Input Parameter

Input Parameters	Description
tag_name	tag name
tweet_id	tweet id that contain that tag

Output Parameter

Output Parameters	Description
none	none

Class

User

Attributes	Description
id	user id
name	user name
email	user email
gender	user gender
birthday	user birthday

Tweet

Attributes	Description
id	tweet id
user_id	the owner of the tweet
content	the content of the tweet
likes	the number of the users who like this tweet
reply_to_tweet_id	If this tweet is a reply, this attribute points to the original tweet. Otherwise it is null

Notification

Attributes	Description
id	notification id
target_user_id	who this notification sends to
type	“mention” , “reply” , “new_follower”
status	“read” ,” unread”
tweet_id	if it is a mention or reply, tweet_id points to that tweet, if it is not, then tweet_id equals null
new_follower_id	if type is a “new_follower” , this points to the follower id, otherwise, it is null