Project 2 API

Jason Otter, Eric van der Roest, Jordan Wermuth

Contents:

NOTE: Any HTTP methods other than GET should have the header Content-Type header set to application/json.

Directory:

Route: /api/
Supported Methods: GET

This endpoint is available in **both** the **users** and **timelines** service and will return a directory of the resources available in the API, but not all methods will be available. The methods available will depend on the service that is being run.

Users Service:

This service contains the endpoints for creating users/followers, password authentication, and deleting followers. There are also some methods for easy testing to retrieve accounts and followers from the database.

Users Service Endpoints:

Route: /api/accounts
Supported Methods: POST, PUT

POST - ***createUser(username, email, password)***

This method inserts a new record into the users table. The request body should include the arguments in the following format:

```
{
	“username”: “new_username”,
	“email”: “new_email”,
	“password”: “new_password”
}
```

This should return a **201** on success, or **409** if the user already exists.

PUT - ***authenticateUser(username, password)***

This method takes a username and password combination and the given password is correct. The request body should include the arguments in the following format:

```
{
	“username”: ”some_username”,
	“password”: “some_password”
}
```

If the supplied password is incorrect, this will return a **401**, otherwise it will return a **200** on successful authentication.

Route: /api/followers
Supported Methods: GET, POST

POST - ***addFollower(username, usernameToFollow)***

This method takes two usernames as arguments, username specifies who the new follower is, and usernameToFollow is the account which they are following. The arguments should be in the following format:

```
{
	“username”: “some_username”,
	“usernameToFollow”: “some_other_username”
}
```

This method should return a **201** on success, or a **409** if the user is already following the specified account. It may also return a **400** if something is wrong with the request (ex: user trying to follow themselves)

Route: /api/accounts/<arg: usernameToRemove>/followers/<arg: username>

Supported Methods: DELETE

DELETE - ***deleteFollower(username, usernameToRemove)***

This method takes as arguments a pair of usernames, the **key difference** is that this method takes **url parameters**. It will then delete the specified pair from the followers table. The arguments should be in the following format:

```
{
        “username”: “some_username”,
        “usernameToRemove”: “some_other_username”
}
```

Should return a **200** on success, or a **404** if the record was not found, or already deleted.

Timelines Service

Route: /api/tweets
Supported Methods: GET, POST

GET - ***getPublicTimeline()***

This method returns the 25 most recent tweets, with the latest appearing first. It takes no arguments, and returns the data as a JSON object.

Should return a **200** on success with the relevant data.

POST - ***postTweet(username, text)***

This method inserts a new tweet into the database. It takes two arguments, the author of the tweet, and the content of the message. The arguments should have the following form:

```
{
        “username”: “some_username”,
        “text”: “tweet_text”
}
```

Should return a **201** on success.

Route: /api/tweets/<arg: author>
Supported Methods: GET

GET - ***getUserTimeline(author)***

This method returns the 25 most recent tweets from the specified author with the most recent tweets first. This method takes as input a **url parameter** specifying the author.

Should return **200** on success with the relevant data, or a **404** if the author has not posted any tweets.

Route: /api/followers/<arg: username>/tweets
Supported Methods: GET

GET - ***getHomeTimeline(username)***

This method returns the 25 most recent tweets from all users the given username follows. This method takes as input a username in the form of a **url parameter**.

Should return **200** on success with the relevant data, or a **404** if the account does not exist.