**Mini Twitter API**

* *An endpoint to read the message list for the current user (as identified by their HTTP Basic authentication credentials). Include messages they have sent and messages sent by users they follow. Support a “search=” parameter that can be used to further filter messages based on keyword.* 
  + Http Basic Authentication Credentials:

USERS:

* + - Username: 1 Password: password1
    - Username: 2 Password: password2

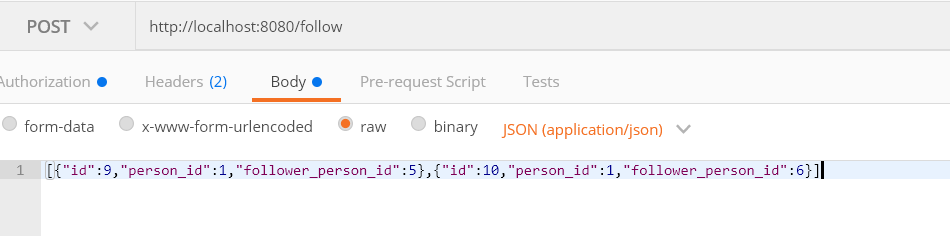
ADMIN:

Username: admin1 Password: secret1

**Note**: Delete cookie and Cache on browser to delete cached Http Basic credentials while using subsequent requests or use Postman to send requests

* + <http://localhost:8080/messages> (List all my Messages and messages of my followers)-: *GET Request implemented using Basic Authentication.*
  + http://localhost:8080/mymessages?search=Intelligence is a privilege, and it needs to be used for the greater good of people. (Search functionality for a user)
* *Endpoints to get the list of people the user is following as well as the followers of the user.* 
  + http://localhost:8080/followers (GET Request) Authentication needed
* *An endpoint to start following another user.* 
  + <http://localhost:8080/follow> (POST Request ) User is authenticated and based on his credentials he is allowed to follow other users. A user can send a list of Followers in the body of the POST request along with the authentication headers for following other users. A User cannot make another user follow someother user. He can only do for his profile.

Sample Body of the Request:



* *An endpoint to unfollow another user.* (POST Request )
  + <http://localhost:8080/unfollow>

User is authenticated and based on his credentials he is allowed to Unfollow other users. Body of the request is similar to follow request.

* *An endpoint to search all the messages posted on the app- ADMIN Funcionality*

<http://localhost:8080/searchmessages?search=sed>

**Notable Features:**

1. Bean validation has been done suing JSR 303 Specification: <http://beanvalidation.org/1.0/spec/>
2. Used the full text search feature provided by H2 database which internally Uses the Apache Lucene Fulltext Search. This functionality has been provided for the admin where he can do a text search based on partial texts.

**Future Upgrades:**

1. User Authentication with OAuth 2.0
2. Utilizing a SQL database like MYSQL, DB2 to store data and also deploying on a JBOSS or Websphere Application Server.
3. Implementing as many Design patterns as possible which suits the design

**Instructions for Testing: The below instructions would help to run a Load Test on the end point URL’s using Apache Apache Jmeter 3.2 (Open source Tool)**

* After downloading the Apache Jmeter , go to the Bin folder and run the ApacheJMeter.jar file. Select file open and point it to the jmeter script(.jmx) file to open the script.
* The Play button would run all the services with a pre-configured parameters.
* The Authorization Manager for each requests can be changed with the required username/password to get info for the required user.
* The parameters can be changed in the Body/Request to make changes to the request w.r.t User/Body Data.
* A couple of Listeners have been added that would give information like Response Time for each **Request/Total Time/Average Time/ Throughput/ Received Kb/sec**.
* The above thread group can be changed to reflect a load testing scenario to check the response time in a prod env with predefined concurrent users.

