Twitter APIs

I have found 2 different twitter APIs for Java: Twitter4J and HBC.

Twitter4J allows me to post a tweet:

public String createTweet(String tweet) throws TwitterException{

Twitter twitter = getTwitterinstance();

    Status status = twitter.updateStatus("creating baeldung API");

    return status.getText();

}

Twitter4J also allows me to fetch a list of tweets from the user’s timeline:

public List<String> getTimeLine() throws TwitterException {

    Twitter twitter = getTwitterinstance();

    return twitter.getHomeTimeline().stream()

      .map(item -> item.getText())

      .collect(Collectors.toList());

}

I can also send a dirrect message:

public static String sendDirectMessage(String recipientName, String msg)

  throws TwitterException {

    Twitter twitter = getTwitterinstance();

    DirectMessage message = twitter.sendDirectMessage(recipientName, msg);

    return message.getText();

}

As well as search for tweets

public static List<String> searchtweets() throws TwitterException {

    Twitter twitter = getTwitterinstance();

    Query query = new Query("source:twitter4j baeldung");

    QueryResult result = twitter.search(query);

    return result.getTweets().stream()

      .map(item -> item.getText())

      .collect(Collectors.toList());

}

The following code in HBC helps in retrieving user tweets

public class FilterStreamExample {

public static final String CONSUMER\_KEY = "Your\_consumer\_key";

public static final String CONSUMER\_SECRET = "Your\_consumer\_secret";

public static final String ACCESS\_TOKEN = "Your\_access\_token";

public static final String ACCESS\_TOKEN\_SECRET = "Your\_access\_token\_secret";

public static void process(String consumerKey, String consumerSecret,

String token, String secret) throws InterruptedException {

//

// Create queue which would be used to get message

//

BlockingQueue queue = new LinkedBlockingQueue(10000);

//

// Create an endpoint of type StatusesFilterEndpoint; It has APIs to retrieve

// users tweets or treats related with mention or hashtags

//

StatusesFilterEndpoint endpoint = new StatusesFilterEndpoint();

//

// Add one or more users to follow the tweets;

//

endpoint.followings(Lists.newArrayList( 136976940L));

//

// Create OAuth object using consumer keys/secret and access token/secret

//

Authentication auth = new OAuth1(consumerKey, consumerSecret, token,

secret);

//

// Create a new BasicClient. By default gzip is enabled.

//

Client client = new ClientBuilder().hosts(Constants.STREAM\_HOST)

.endpoint(endpoint).authentication(auth)

.processor(new StringDelimitedProcessor(queue)).build();

//

// Establish a connection

//

client.connect();

//

// Code below would extract message as it appears on Twitter

// Do whatever needs to be done with messages; In the code below,

// the message is printed; In real world, the message could be stored

// in Hadoop storage

//

for (int msgRead = 0; msgRead < 1000; msgRead++) {

String msg = queue.take();

System.out.println(msg);

}

client.stop();

}

public static void main(String[] args) {

try {

FilterStreamExample.process(CONSUMER\_KEY, CONSUMER\_SECRET,

ACCESS\_TOKEN, ACCESS\_TOKEN\_SECRET);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

The following details are required to get going. These details could be obtained from twitter developers website. Same applies for Twitter4J.

* Consumer Key
* Consumer Secret
* Access Token
* Access Token Secret
* In order to get the tweets as they appear for certain users, the userId is required.
* StatusesFilterEndpoint class is used to get the user tweets. It has this API “public StatusesFilterEndpoint followings(List<Long> userIds)” which makes it happen.
* A ClientBuilder is used to create a Client object using following information:
  + OAuth object created using consumer and access token details
  + Host information
  + Endpoint information
  + Queue object which would be used to deliver message to this program