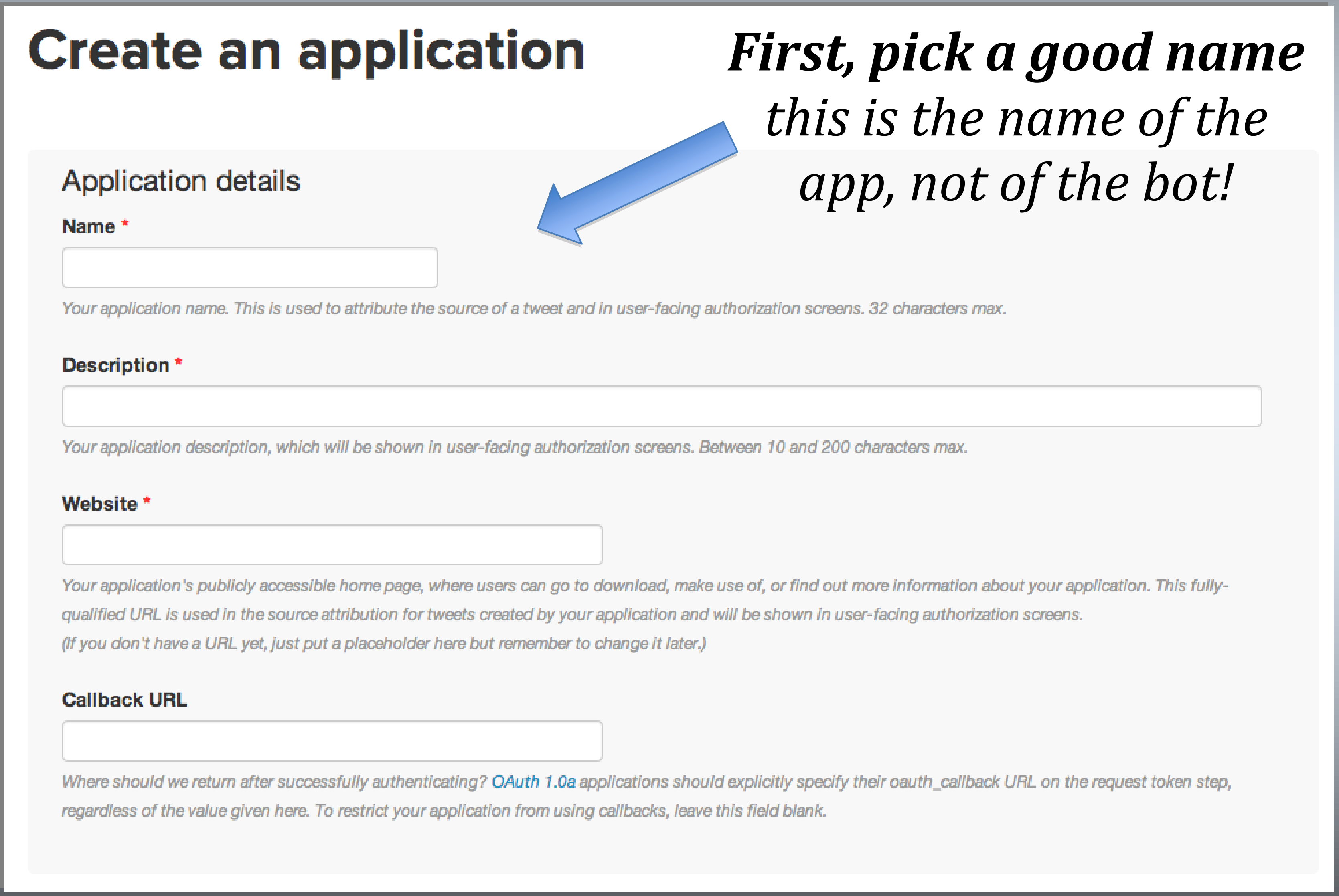
So *dummy*, I am going to configure you to "*tweet*" for me. I am going to set you up as a Twitterbot
Hmmm ... Can I generate ***Evil*** *Tweets?*

How to Train Your Twitterbot: Part I

Go this set to register your app:

/apps.twitter.com/app/new



Create an application

First, pick a good name

this is the name of the app, not of the bot!

Application details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

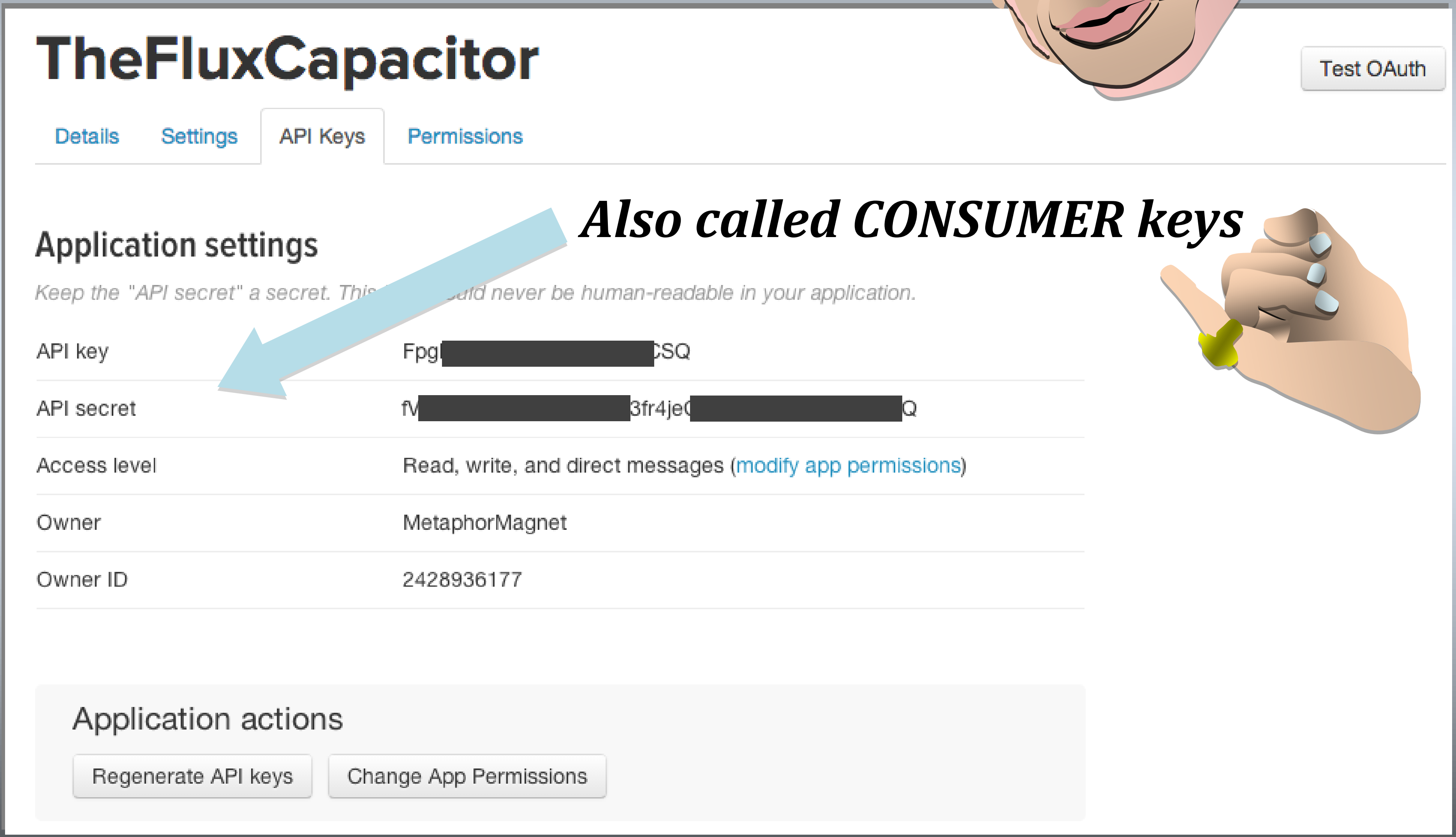
Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

The app behind @*MetaphorMagnet* is called: **FluxCapacitor**



TheFluxCapacitor

Also called CONSUMER keys

Application settings

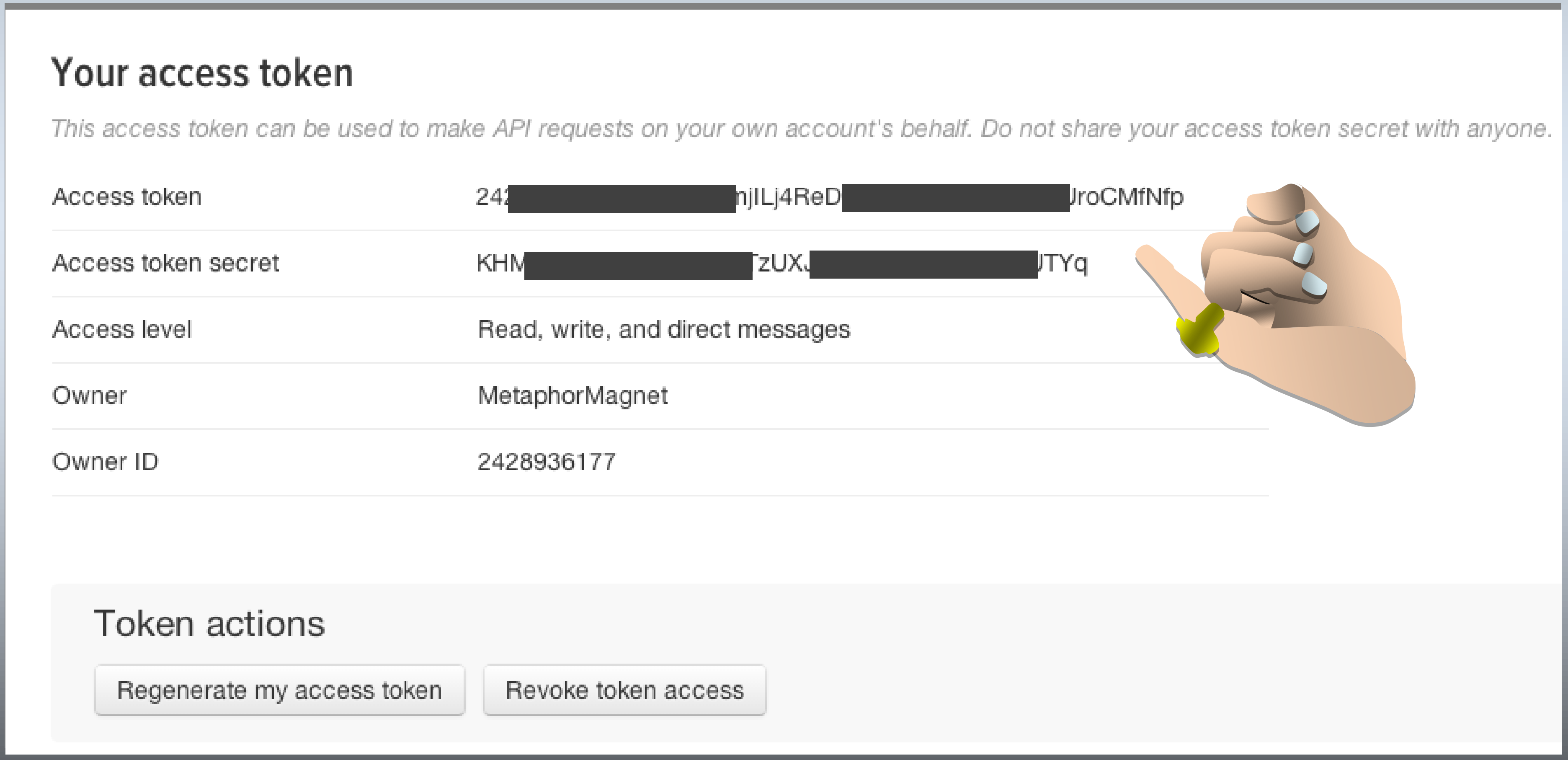
Keep the "API secret" a secret. This [illegible] uld never be human-readable in your application.

API key	Fpg[redacted]CSQ
API secret	fV[redacted]3fr4je[redacted]Q
Access level	Read, write, and direct messages (modify app permissions)
Owner	MetaphorMagnet
Owner ID	2428936177

Application actions

You will also get private access tokens:

Keep these **A Frickin' Secret!**



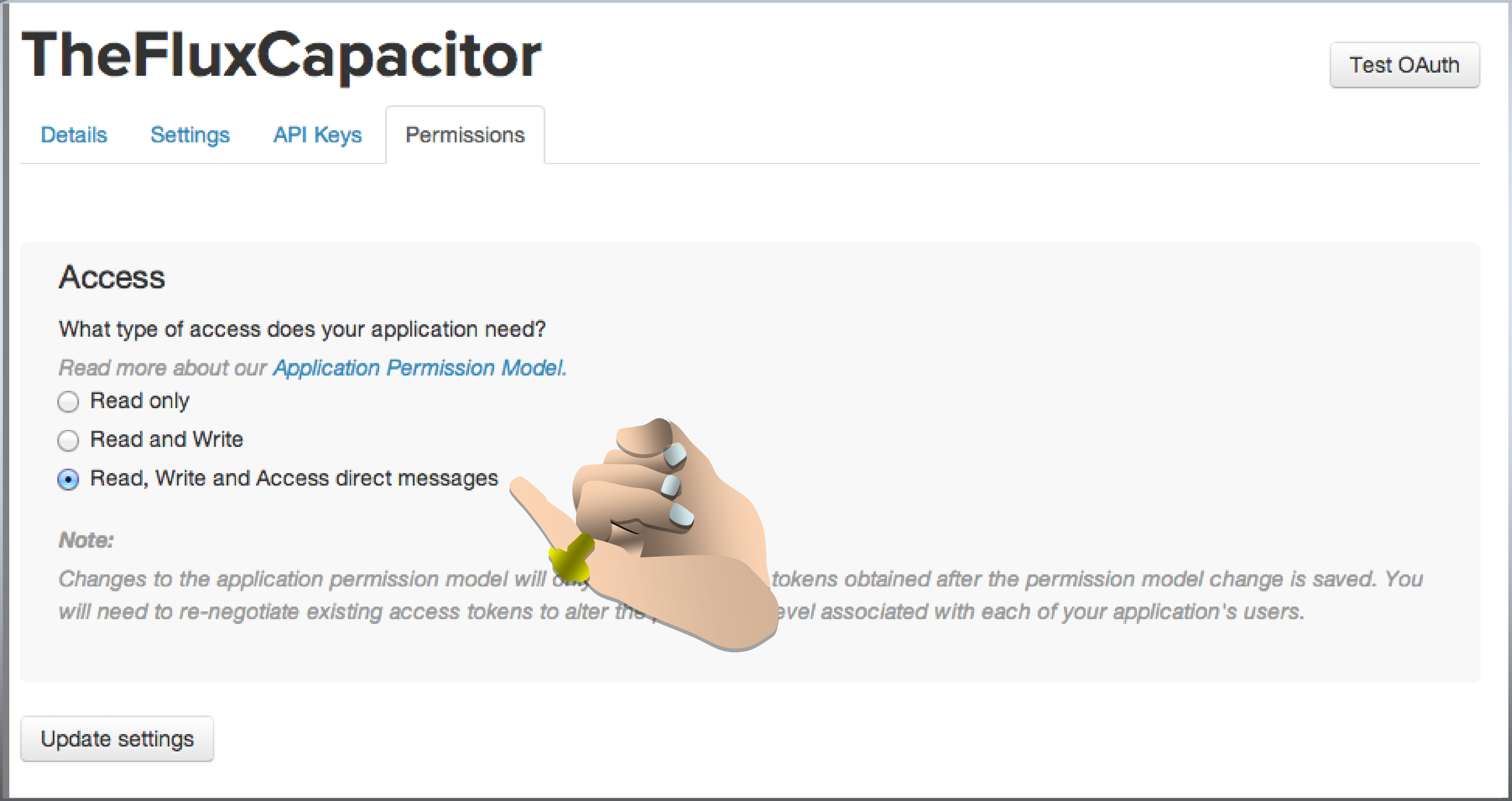
Your access token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access token	24[redacted]jILj4ReD[redacted]roCMfNfp
Access token secret	KH[redacted]zUX[redacted]TYq
Access level	Read, write, and direct messages
Owner	MetaphorMagnet
Owner ID	2428936177

Token actions

Finally, configure the settings for your app: You want to: **Read & Write**



TheFluxCapacitor

Access

What type of access does your application need?

Read more about our Application Permission Model.

○ Read only

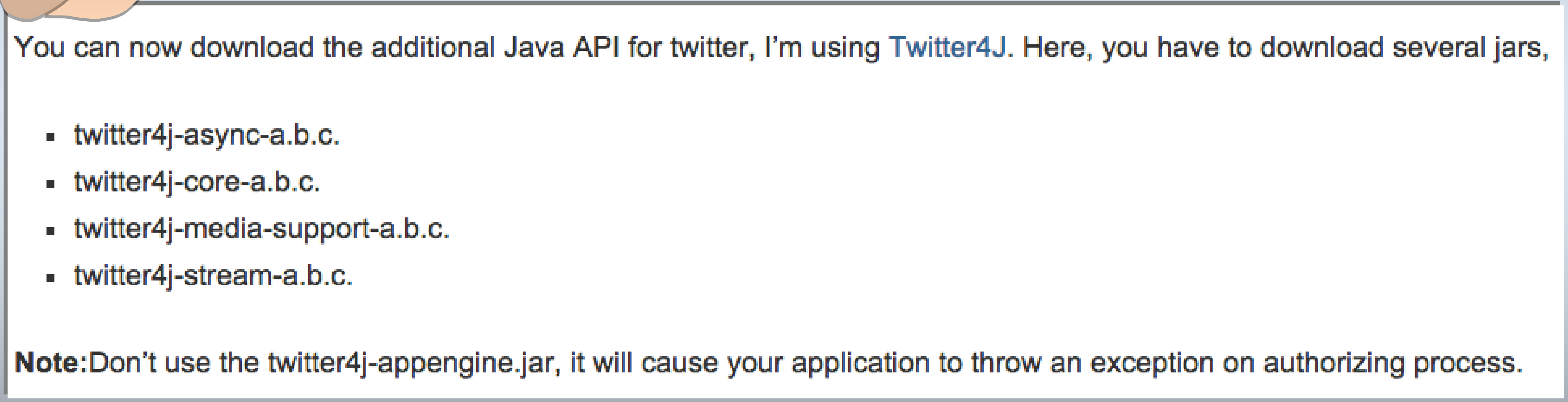
○ Read and Write

● Read, Write and Access direct messages

Note:

Changes to the application permission model will o tokens obtained after the permission model change is saved. You will need to re-negotiate existing access tokens to alter the p evel associated with each of your application's users.

Now, we will access our app via a Java API for Twitter called: **Twitter4J**



You can now download the additional Java API for twitter, I'm using Twitter4J. Here, you have to download several jars,

- twitter4j-async-a.b.c.
- twitter4j-core-a.b.c.
- twitter4j-media-support-a.b.c.
- twitter4j-stream-a.b.c.

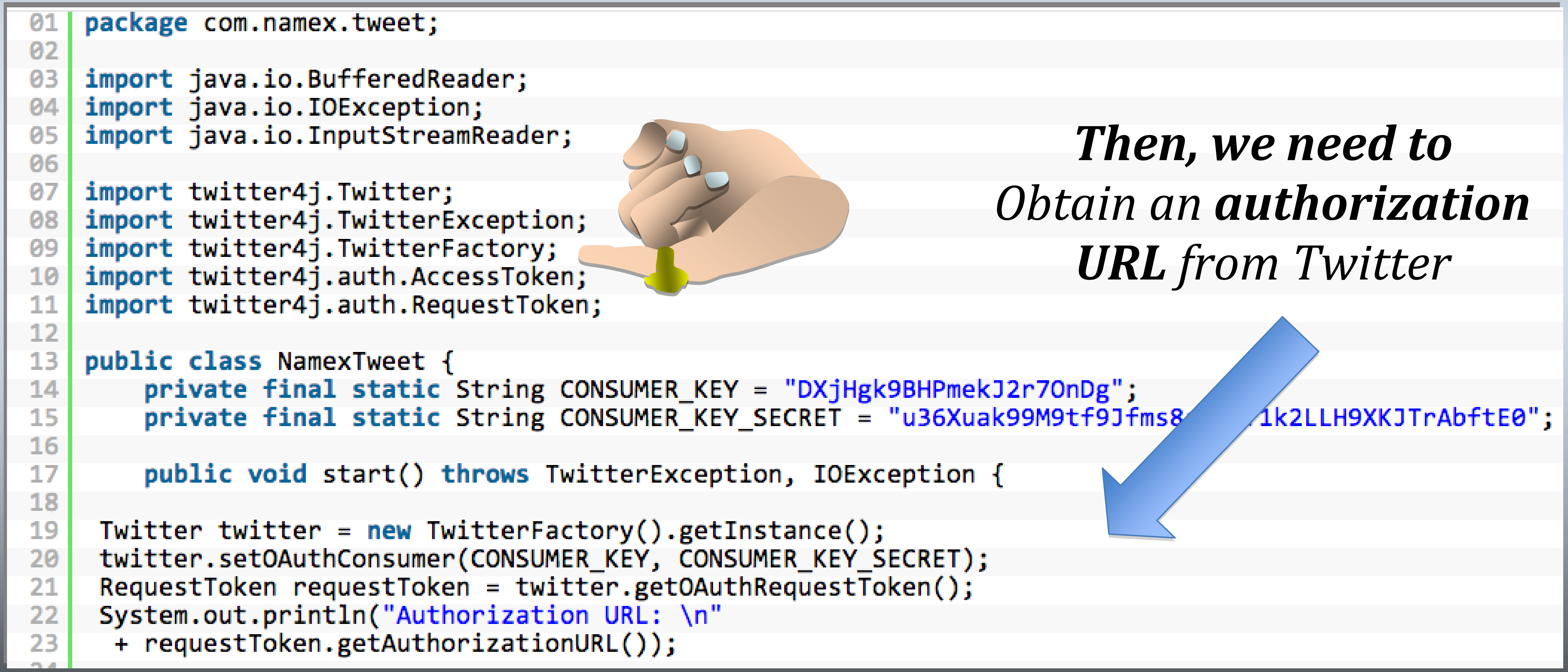
Note: Don't use the twitter4j-appengine.jar, it will cause your application to throw an exception on authorizing process.

From:

http://www.javacodegeeks.com/2011/10/java-twitter-client-with-twitter4j.html

Now, let’s write some frickin’ Java:

Import the API classes: **Twitter4j**

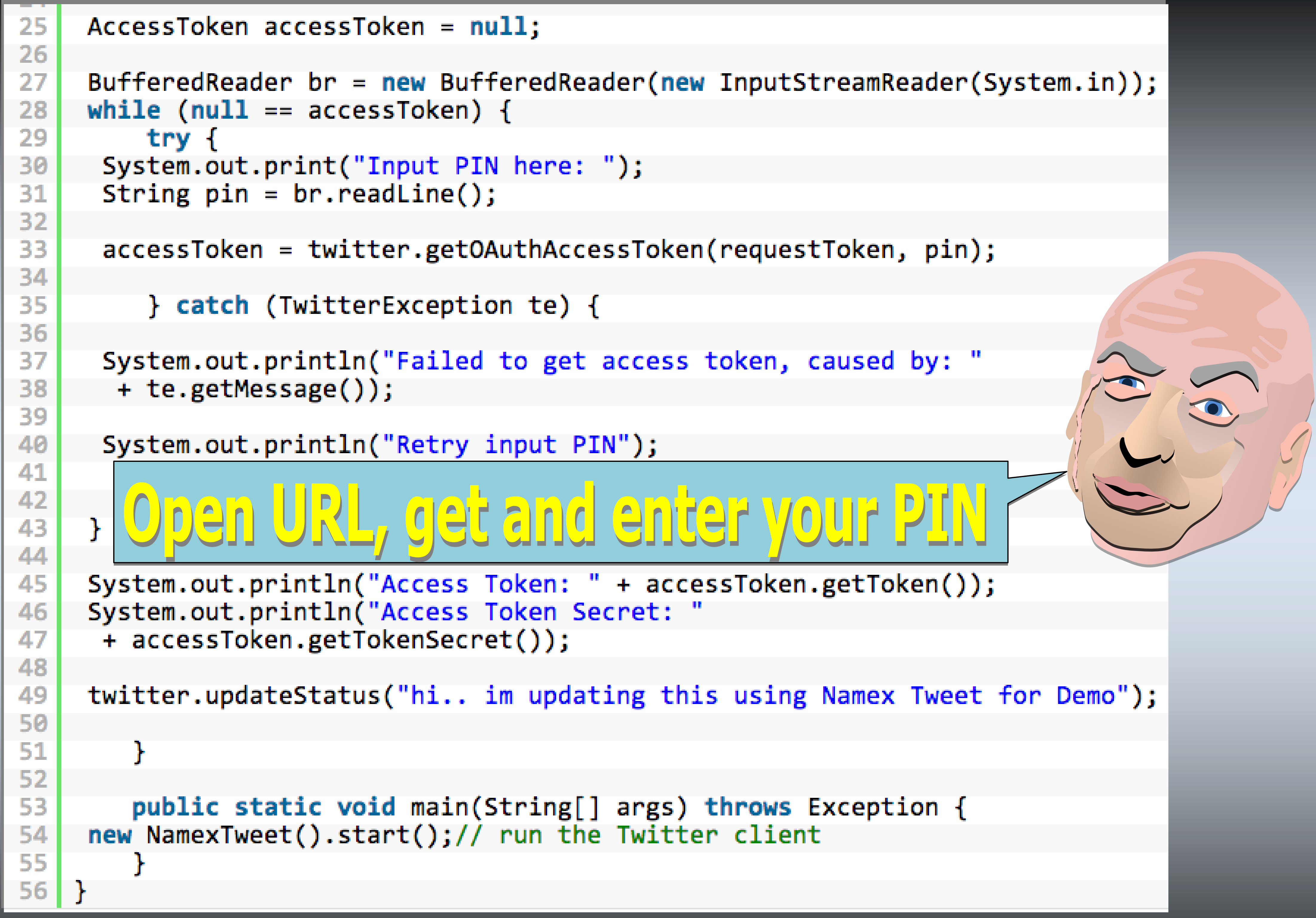


```
package com.namex.tweet;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.auth.AccessToken;
import twitter4j.auth.RequestToken;

public class NamexTweet {
    private final static String CONSUMER_KEY = "DXjHgk9BHPmekJ2r7OnDg";
    private final static String CONSUMER_KEY_SECRET = "u36Xuak99M9tf9Jfms8    1k2LLH9XKJTrAbftE0";

    public void start() throws TwitterException, IOException {

 Twitter twitter = new TwitterFactory().getInstance();
 twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_KEY_SECRET);
 RequestToken requestToken = twitter.getOAuthRequestToken();
 System.out.println("Authorization URL: \n"
  + requestToken.getAuthorizationURL());
```

Then, we need to

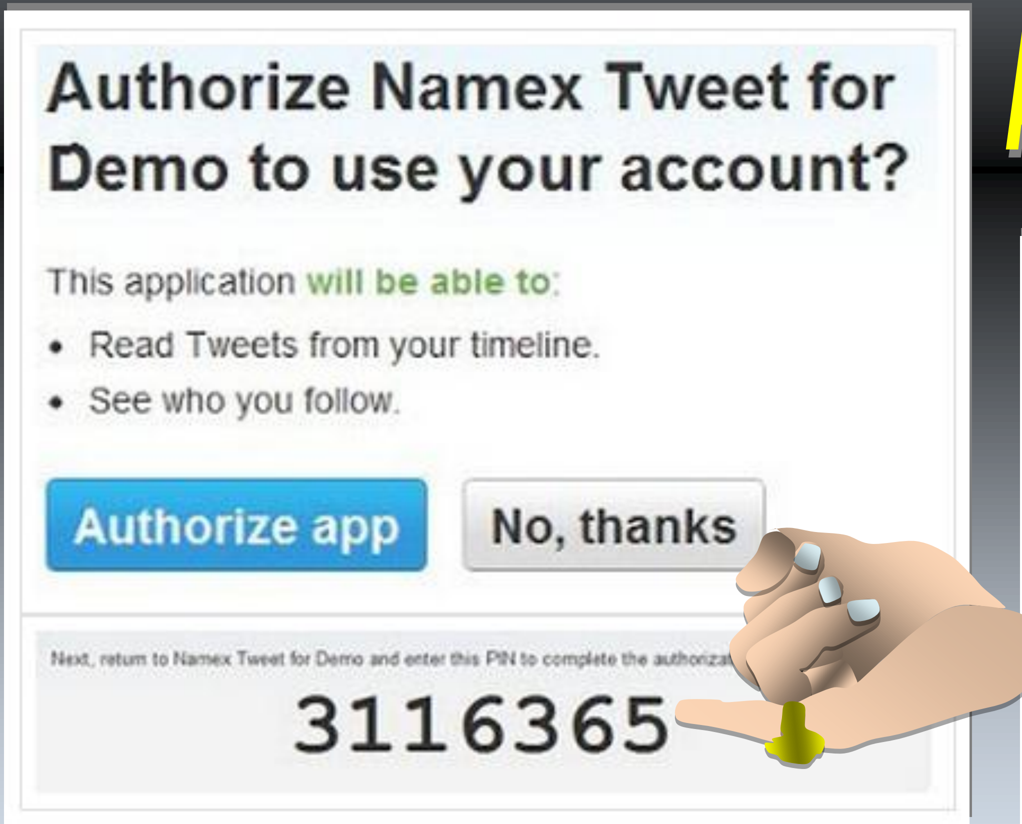
Obtain an ***authorization***

URL *from Twitter*



```java
 AccessToken accessToken = null;

 BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
 while (null == accessToken) {
     try {
  System.out.print("Input PIN here: ");
  String pin = br.readLine();

  accessToken = twitter.getOAuthAccessToken(requestToken, pin);

     } catch (TwitterException te) {

  System.out.println("Failed to get access token, caused by: "
   + te.getMessage());

  System.out.println("Retry input PIN");


 }

 System.out.println("Access Token: " + accessToken.getToken());
 System.out.println("Access Token Secret: "
  + accessToken.getTokenSecret());

 twitter.updateStatus("hi.. im updating this using Namex Tweet for Demo");

    }

    public static void main(String[] args) throws Exception {
 new NamexTweet().start();// run the Twitter client
    }
}
```

Here's the Authorization page, and here is the pin you receive: **Success!**



Authorize Namex Tweet for Demo to use your account?

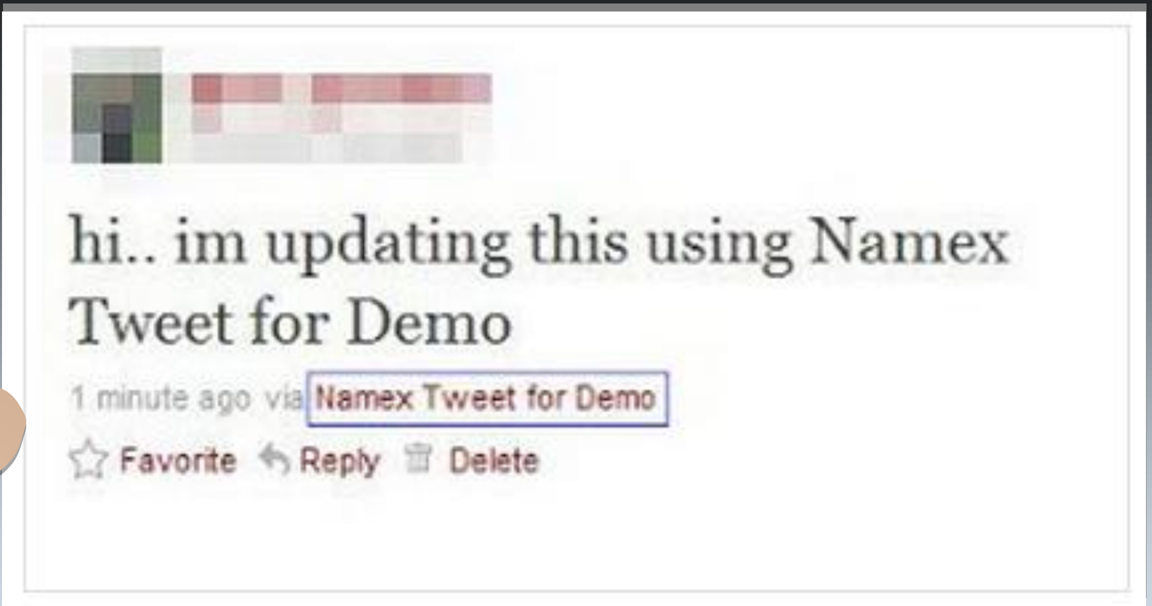
This application will be able to:

- Read Tweets from your timeline.
- See who you follow.

Next, return to Namex Tweet for Demo and enter this PIN to complete the authoriza

3116365

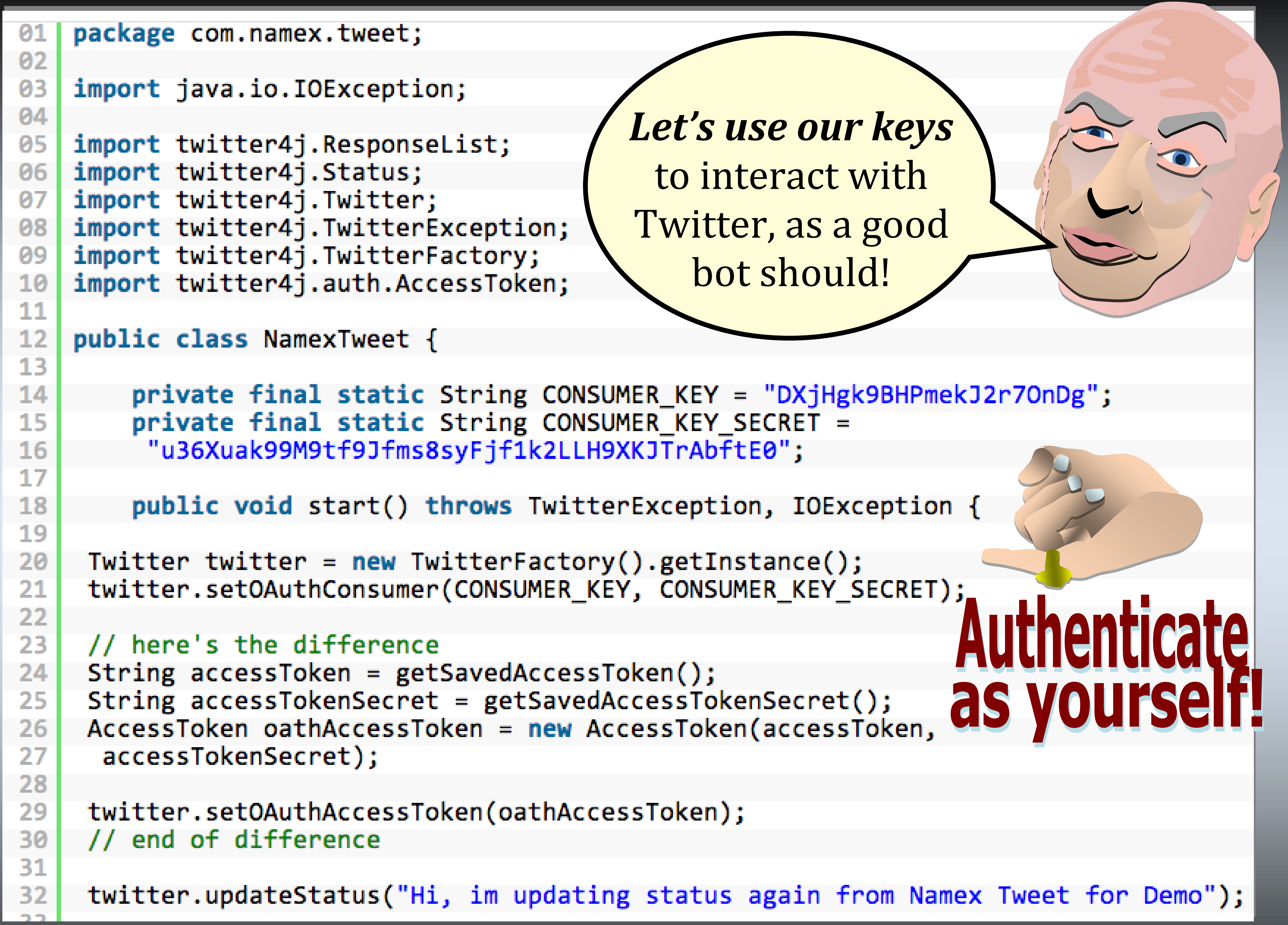
And the first tweet from your app:



[redacted]

hi.. im updating this using Namex Tweet for Demo

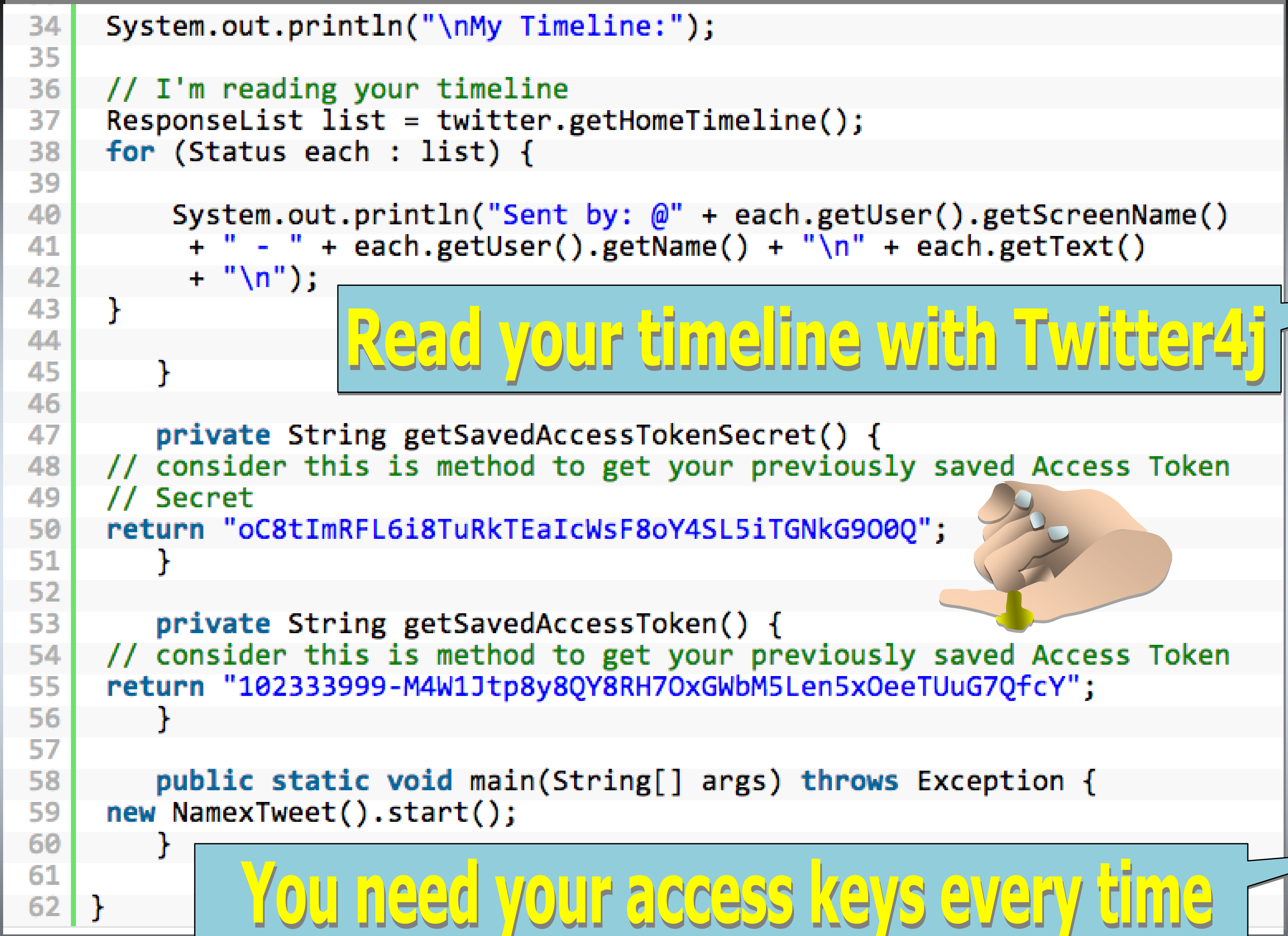
1 minute ago via Namex Tweet for Demo



```java
package com.namex.tweet;

import java.io.IOException;

import twitter4j.ResponseList;
import twitter4j.Status;
import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.auth.AccessToken;

public class NamexTweet {

    private final static String CONSUMER_KEY = "DXjHgk9BHPmekJ2r7OnDg";
    private final static String CONSUMER_KEY_SECRET =
     "u36Xuak99M9tf9Jfms8syFjf1k2LLH9XKJTrAbftE0";

    public void start() throws TwitterException, IOException {

 Twitter twitter = new TwitterFactory().getInstance();
 twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_KEY_SECRET);

 // here's the difference
 String accessToken = getSavedAccessToken();
 String accessTokenSecret = getSavedAccessTokenSecret();
 AccessToken oathAccessToken = new AccessToken(accessToken,
  accessTokenSecret);

 twitter.setOAuthAccessToken(oathAccessToken);
 // end of difference

 twitter.updateStatus("Hi, im updating status again from Namex Tweet for Demo");
```

Let's use our keys to interact with Twitter, as a good bot should!

Authenticate as yourself!



```
 System.out.println("\nMy Timeline:");

 // I'm reading your timeline
 ResponseList list = twitter.getHomeTimeline();
 for (Status each : list) {

     System.out.println("Sent by: @" + each.getUser().getScreenName()
      + " - " + each.getUser().getName() + "\n" + each.getText()
      + "\n");
 }

    }

    private String getSavedAccessTokenSecret() {
 // consider this is method to get your previously saved Access Token
 // Secret
 return "oC8tImRFL6i8TuRkTEaIcWsF8oY4SL5iTGNkG9O0Q";
    }

    private String getSavedAccessToken() {
 // consider this is method to get your previously saved Access Token
 return "102333999-M4W1Jtp8y8QY8RH7OxGWbM5Len5xOeeTUuG7QfcY";
    }

    public static void main(String[] args) throws Exception {
 new NamexTweet().start();
    }

}
```

Read your timeline with Twitter4j

You need your access keys every time

Your bot is now ready to be unleashed on the world!

Mwah-hahahaha