```
//author(s): Jimmie Potts, Alan Birmaher
package com.cyf.challengeyourfriends;

import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.auth.AccessToken;
import twitter4j.auth.RequestToken;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;

import com.hintdesk.core.util.StringUtil;



public class TwitterActivity extends Activity {

    private final Context theC = this;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        initControl();
    }

    private void initControl() {
        final Uri uri = getIntent().getData();
        if (uri != null && uri.toString().startsWith(ConstantValues.
            TWITTER_CALLBACK_URL)) {
            final String verifier = uri.getQueryParameter(ConstantValues.
                URL_PARAMETER_TWITTER_OAUTH_VERIFIER);
            new TwitterGetAccessTokenTask(this).execute(verifier);
        } else
            new TwitterGetAccessTokenTask(this).execute("");
    }

    class TwitterGetAccessTokenTask extends AsyncTask<String, String, String> {

        private Context context;
        public TwitterGetAccessTokenTask(Context context) {
            this.context = context;
        }
        @Override
        protected void onPostExecute(String userName) {
             Intent intent = new Intent(context, MenuActivity.class);
             startActivity(intent);
        }

        @Override
        protected String doInBackground(String... params) {

            final Twitter twitter = TwitterUtil.getInstance().getTwitter();
            final RequestToken requestToken = TwitterUtil.getInstance().
                getRequestToken();
            if (!StringUtil.isNullOrWhitespace(params[0])) {
                try {
                    final AccessToken accessToken = twitter.getOAuthAccessToken
                        (requestToken, params[0]);
                    final SharedPreferences sharedPreferences = PreferenceManager
                        .getDefaultSharedPreferences(theC);
                    final SharedPreferences.Editor editor = sharedPreferences.
                        edit();
                    editor.putString(ConstantValues.
                        PREFERENCE_TWITTER_OAUTH_TOKEN, accessToken.getToken());
                    editor.putString(ConstantValues.
                        PREFERENCE_TWITTER_OAUTH_TOKEN_SECRET, accessToken.
                        getTokenSecret());
                    editor.putBoolean(ConstantValues.
                        PREFERENCE_TWITTER_IS_LOGGED_IN, true);
                    editor.commit();
                    return twitter.showUser(accessToken.getUserId()).getName();
                } catch (TwitterException e) {
                    e.printStackTrace();
                }
            } else {
                final SharedPreferences sharedPreferences = PreferenceManager.
                    getDefaultSharedPreferences(theC);
                final String accessTokenString = sharedPreferences.getString
                    (ConstantValues.PREFERENCE_TWITTER_OAUTH_TOKEN, "");
                final String accessTokenSecret = sharedPreferences.getString
                    (ConstantValues.PREFERENCE_TWITTER_OAUTH_TOKEN_SECRET, "");
                final AccessToken accessToken = new AccessToken(accessTokenString
                    , accessTokenSecret);
                try {
                    TwitterUtil.getInstance().setTwitterFactory(accessToken);
                    return TwitterUtil.getInstance().getTwitter().showUser
                        (accessToken.getUserId()).getName();
                } catch (TwitterException e) {
                    e.printStackTrace();
                }
            }
            return null;
        }
    }
}
```