

Authentication Using Oauth 2 for Twitter and Instagram REST calls

J J Martin

June 24, 2013

1 Introduction

To connect to most social media sites, they require you to authenticate the user and/or application that wants to connect to the users behalf to show and modify the status and posts on there profile. Before any REST calls can be made to one of these websites you are required to authenticate with the social media server most commonly Oauth is used.

2 Twitter Oauth2

Oauth 2 is used for Twitters Application-only authorization. It is a 3 phase process to allow a application authorization to search through the requested tweets. Once an application is registered on Twitter it will be given a unique client-id and client-secret. These are the two distinct values that represent your application to Twitter which is used to generate access tokens for you application.

2.1 Phase 1

The application URL encodes the client-id and client-secret, then concatenates the strings separated by a “:” which then gets encoded by base 64 encoding known as “binary to text” encoding.

2.2 Phase 2

The application then makes a “POST” request to the authentication server “api.twitter.com/oauth2/token” using Https (port 443) with a header attribute called Authorization and value of “Basic base64_String” where base64_String is the binary to text encoded string in phase 1, and one other attributes in the header “Content-Type” with a value of “application/x-www-form-urlencoded; charset=UTF-8”. The body of the request must be “grant_type=client_credentials”.

2.3 Phase 3

Phase 2 should return with a Http status code 200, with an encrypted payload under the gzip encryption (**be sure to decrypt the payload**). Be sure to check that the “**token_type**” is “bearer” but most importantly you should receive a “**access_token**” this will be used to verify that it is your application that is attempting to access Twitter. Now that an access token has been issued we can make requests to the Twitter API with the key, value pair “Authorization” and “Bearer **access_token**” in the header of each request.