



Using Twitter™ & Facebook™ API

INF5750/9750 - Lecture 10 (Part II)

Lecture contents

- Connecting to popular social APIs
- Authentication
- Authorization
- Common calls
- Privacy and understanding data storage

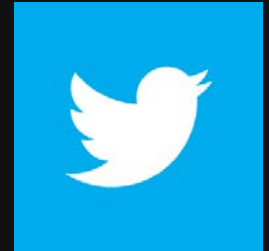
Social APIs

- An API to connect different social networks to exchange data
- Users can interact with:
 - Authorization
 - User Profile
 - Activity Streams/Data/Posts
 - Friends/Followers/Connections...
 - ...
- OpenSocial? – interoperable formats
 - Started with Orkut
 - Few really common resources and formats

Authentication

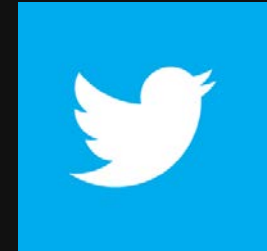
- Multiple authentication schemes before standardizing to OAuth. OpenID gave rise to the idea of OAuth.
- Differences in v1.0 and v2.0, but we'll skip the details
- Few webservices will still allow 2 common authentication schemes:
 - Basic authentication
 - Digest authentication
- Common terms:
 - “OAuth dance” - unofficial term to describe flow
 - Request token - lets service know “what” is requesting access
 - Access token - authorized request token is exchanged for an access token
 - Endpoints - URLs where token are requested and exchanged
 - Service Provider - the provider of the web service
 - Consumer - application that the user will use

Twitter Authorization

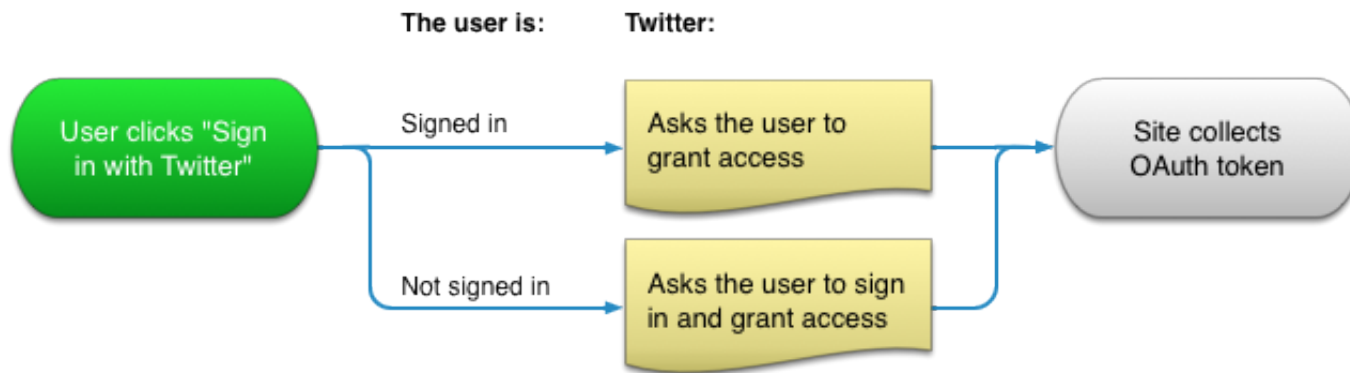


- Broadly in two ways:
 - Application-user authentication
 - Application-only authentication
- Application-user uses OAuth 1.0A
 - Per request
 - Token-based
- Application-only uses OAuth 2.0

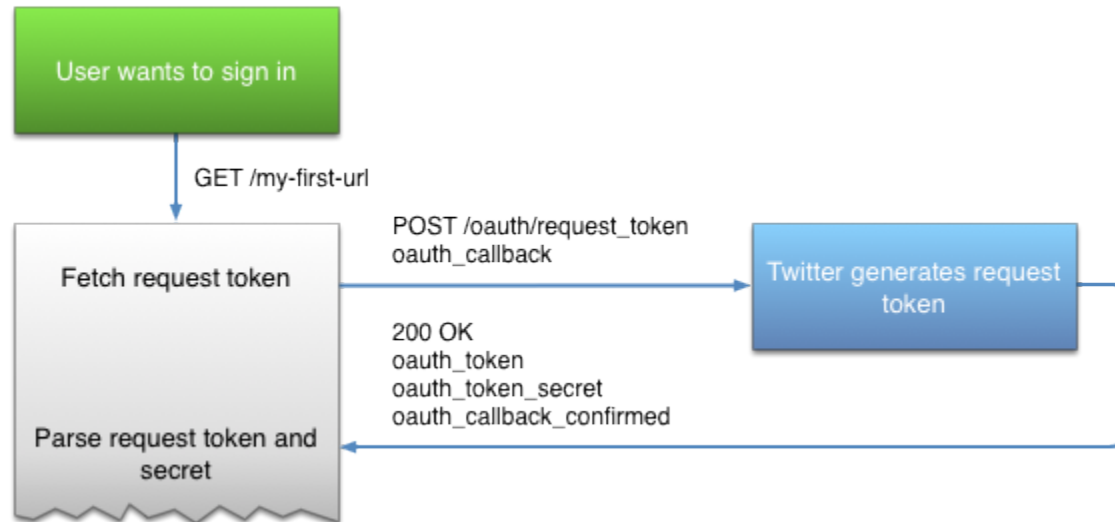
3-legged authorization



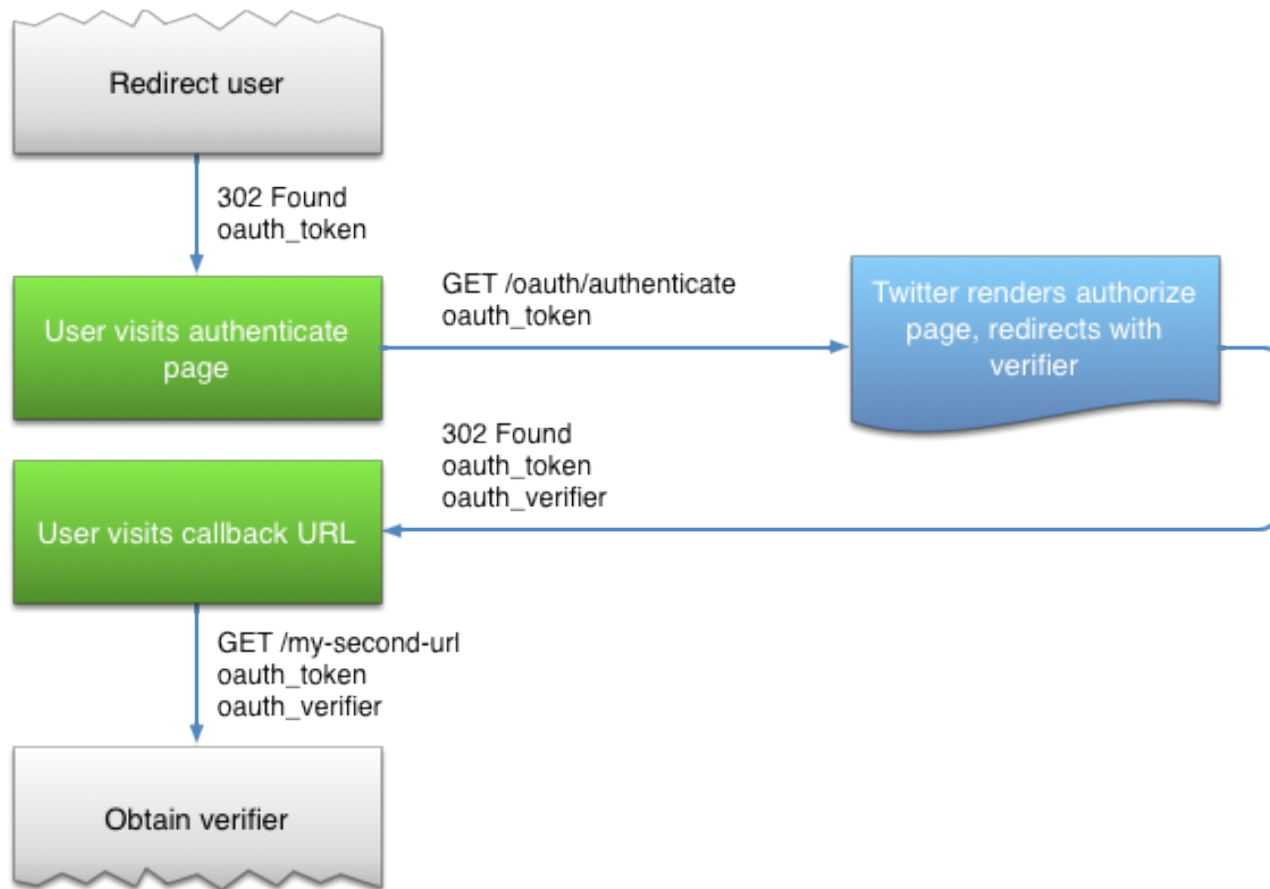
- Authorize application to obtain an **access token** by redirecting a user to Twitter
- [GET oauth/authorize](#) endpoint
- user will **always** be prompted to authorize access to your application, even if access was previously granted.



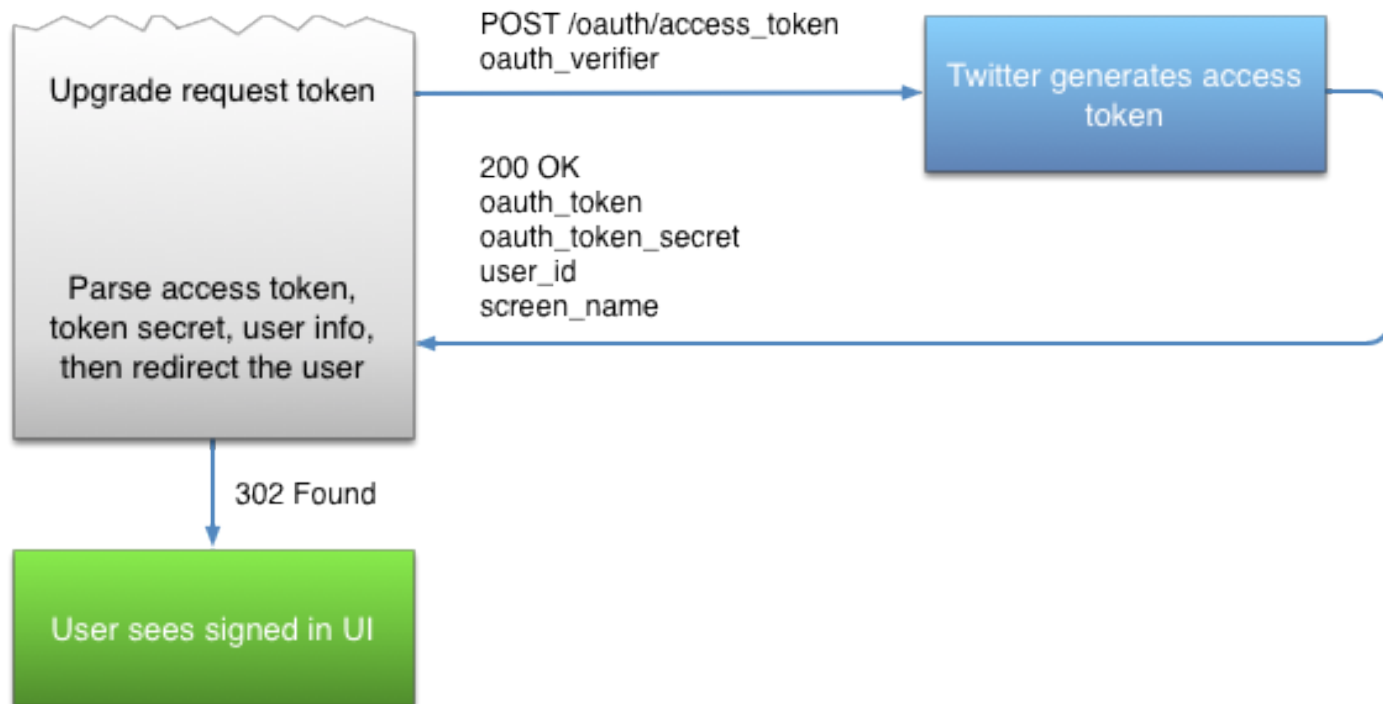
OAuth – Sign in to twitter (Step 1)



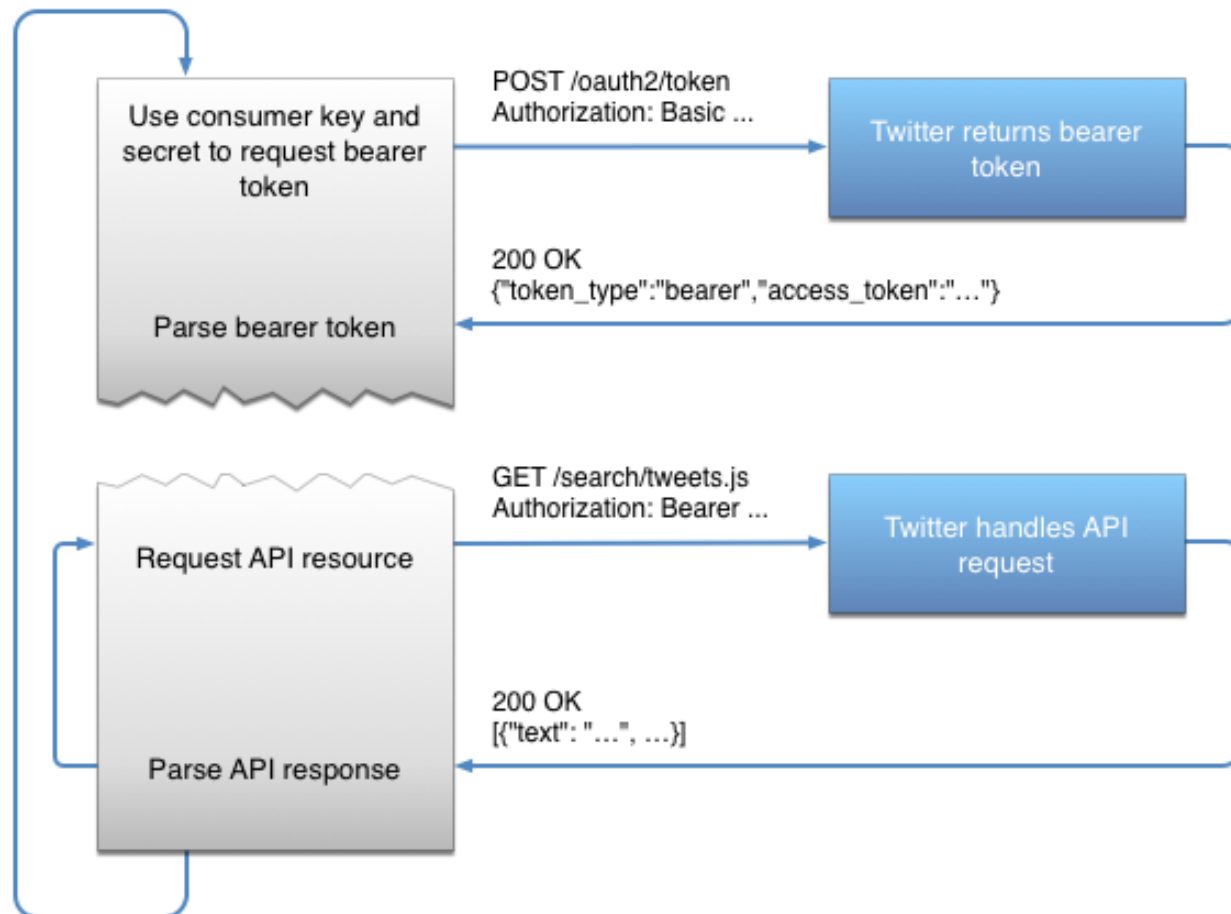
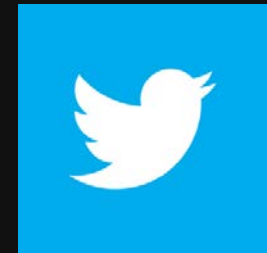
OAuth – Sign in to twitter (Step 2)



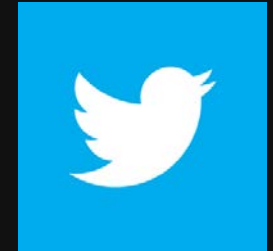
OAuth – Sign in to twitter (Step 3)



Application-only Authentication



Points to remember



- Tokens are passwords
- SSL absolutely required
- No user context
- Rate limiting
 - Separation between calls made on user's behalf compared to those made from apps

- Invalid requests to obtain or revoke bearer tokens

```
{"errors": [{"code": 99, "label": "authenticity_token_error", "message": "Unable to verify your credentials"}]}
```

- API request contains invalid bearer token

```
{"errors": [{"message": "Invalid or expired token", "code": 89}]}
```

- Bearer token used on endpoint which doesn't support application-only auth (e.g. GET statuses/home_timeline)

```
{"errors": [{"message": "Your credentials do not allow access to this resource", "code": 220}]}
```

Service: Authentication: powered by **apigee**

Select an API method

Timelines

- /statuses/mentions_timeline.json
- /statuses/user_timeline.json
- /statuses/home_timeline.json

Tweets

- /statuses/retweets/{id}.json
- /statuses/show/{id}.json
- /statuses/destroy/{id}.json
- /statuses/update.json
- /statuses/retweet/{id}.json
- /statuses/update_with_media.json
- /statuses/oembed.json

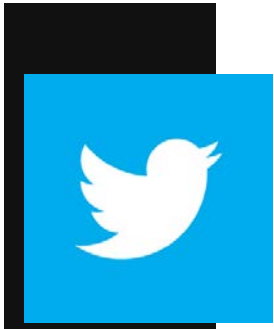
Search

- /search/tweets.json

Help

- /help/configuration.json
- /help/languages.json
- /help/privacy.json
- /help/tos.json
- /application/rate_limit_status.json

response



Twitter Console

<https://dev.twitter.com/console>



Facebook Login Flow

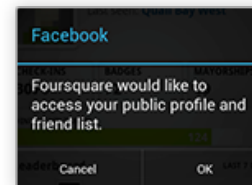
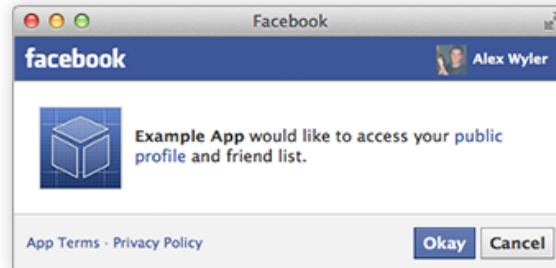
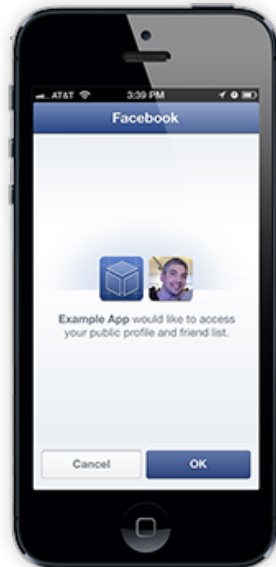
- Easiest is to use the Facebook SDK
- JavaScript SDK with JQuery is for recommended for webapps

```
$(document).ready(function () {
    $.ajaxSetup({ cache: true });
    $.getScript('//connect.facebook.net/en_UK/all.js', function () {
        FB.init({
            appId: 'YOUR_APP_ID',
            channelUrl: '//yourapp.com/channel.html',
        });
        $('#loginbutton,#feedbutton').removeAttr('disabled');
        FB.getLoginStatus(function (response) {
            if (response.status === 'connected') {
                // the user is logged in and has authenticated your app
                var uid = response.authResponse.userID;
                var accessToken = response.authResponse.accessToken;
            } else if (response.status === 'not_authorized') {
                // the user is logged in to Facebook,
                // but has not authenticated your app
            } else {
                // the user isn't logged in to Facebook.
            }
        });
    });
});
```



Login & logout a user

```
FB.login(function (response) {  
  if (response.authResponse) {  
    // The person logged into your app  
  } else {  
    // The person cancelled the login dialog  
  }  
});
```



```
FB.logout(function (response) {  
  // Person is now logged out  
});
```



Facebook APIs

Graph API

The Graph API is a simple HTTP-based API that gives access to the Facebook social graph, uniformly representing objects in the graph and the connections between them. Most other APIs at Facebook are based on the Graph API.

Open Graph

The Open Graph API allows apps to tell stories on Facebook through a structured, strongly typed API.

Dialogs

Facebook offers a number of dialogs for Facebook Login, posting to a person's timeline or sending requests.

Chat

You can integrate Facebook Chat into your Web-based, desktop, or mobile instant messaging products. Your instant messaging client connects to Facebook Chat via the Jabber XMPP service.

Ads API

The Ads API allows you to build your own app as a customized alternative to the Facebook Ads Manager and Power Editor tools.

FQL

Facebook Query Language, or FQL, enables you to use a SQL-style interface to query the data exposed by the Graph API. It provides for some advanced features not available in the Graph API such as using the results of one query in another.

Localization and translation

Facebook supports localization of apps. Read about the tools we provide.

Atlas API

The Atlas APIs provides you with programmatic access to the Atlas web services.

Public Feed API

The Public Feed API lets you read the stream of public comments as they are posted to Facebook.

Keyword Insights API

The Keyword Insights API exposes an analysis layer on top of all Facebook posts that enables you to query aggregate, anonymous insights about people mentioning a certain term.

Privacy and data storage

- Be careful not to store user information without user acceptance and clear guidelines
- User information should have a deletion policy or data withdrawal policy
- User authentication details should not be stored, instead tokens should be stored
- Encryption for tokens should be done in application on device

Resources

- Implementing Twitter Sign-in
 - <https://dev.twitter.com/docs/auth/implementing-sign-twitter>
- Facebook SDK for JavaScript
 - <https://developers.facebook.com/docs/howtos/login/getting-started/>
- Facebook login flow
 - <https://developers.facebook.com/docs/facebook-login/login-flow-for-web>
- Facebook JavaScript SDK
 - <http://developers.facebook.com/docs/reference/javascript/>