



# Open Authorization :

---

OAuth for credentials security in REST API access





Panggi Libersa Jasri Akadol

Web : <http://www.opensecuritylab.org>

Twitter : @panggi



# Agenda

- Web 2.0 and Data
- OAuth usage
- Useful resources



# Web 2.0 and Data



# Web 2.0





= Your Data







Different service = Different data







What if  
you need to use your data that stored in  
another service provider's server ?





Yup , just take it 😊



But How ??



OK .. Enough with the Cute creatures :-p

Let's dive into technical things



Once again.. How ?

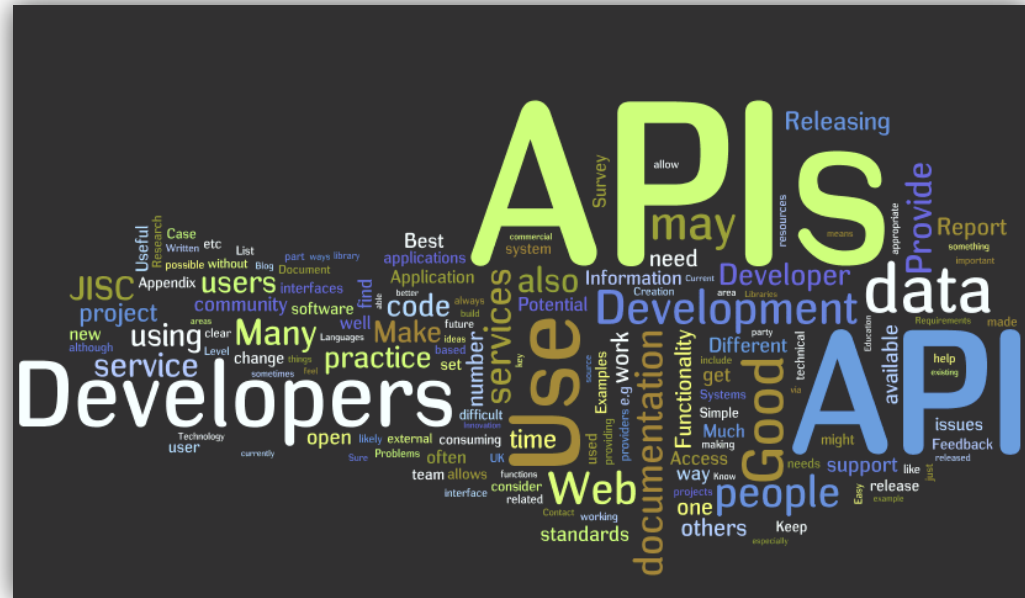


# Using API (Application Programming Interface)



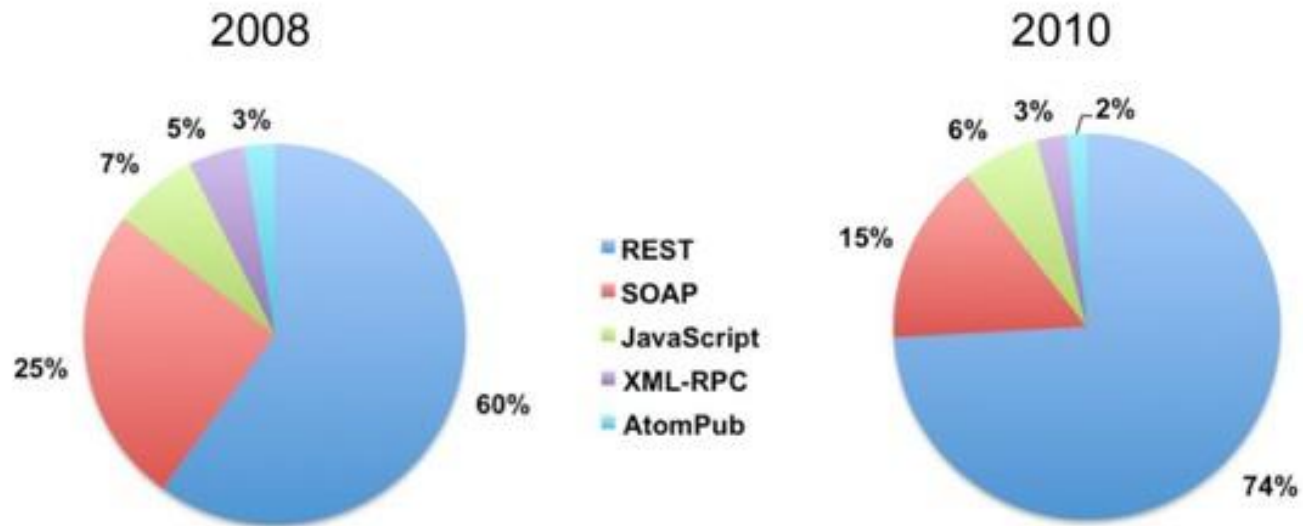


is an interface implemented by a software program that enables it to interact with other software. It facilitates interaction between different software programs similar to the way the user interface facilitates interaction between humans and computers. ( via <http://en.wikipedia.org/wiki/API>)





## REST vs. SOAP: Simplicity wins again



*Distribution of API protocols and styles*

*Based on directory of 2,000 web APIs listed at ProgrammableWeb, May 2010*



# REST

Representational State Transfer

- Provide every resource with a unique ID, for example, a URI
- Link resources with each other, establishing relationships among resources
- Use standard methods (HTTP, media types, XML)
- Resources can have multiple representations that reflect different application states
- The communication should be stateless using the HTTP

## URL

**`http://api.twitter.com/version/statuses/home_timeline.format`**

## Supported formats

**`json, xml, rss, atom`**

## Supported request methods

**`GET`**

## Requires Authentication

**`true`** [About authentication »](#)

## Rate Limited

**`true`** [About rate limiting »](#)

## Parameters

### Optional

- **`since_id`**

Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occurred since the `since_id`, the `since_id` will be forced to the oldest ID available.

`http://api.twitter.com/1/statuses/home_timeline.json?since_id=12345`



# Accessing API

## Find your Network on Spock

Spock can scan your address books and add your contacts to your Favorites on Spock.

- We will not spam your contacts, and will only send them invites to Spock if you request it below.
- We will not sell, display, or otherwise give away your email address.
- We will only login to your services once to discover your contacts on Spock.

## Web Address Book Import

Service	Login	Password
AOL	<input type="text" value="brian@brianoberkirch.c"/> @aol.com	<input type="password" value="*****"/>
LinkedIn	<input type="text" value=""/> (email_address)	<input type="password"/>
Gmail	<input type="text" value=""/> @gmail.com	<input type="password"/>
Hotmail	<input type="text" value=""/> @hotmail.com	<input type="password"/>
Yahoo! Mail	<input type="text" value=""/> @yahoo.com	<input type="password"/>
Plaxo	<input type="text" value=""/> (email_address)	<input type="password"/>

☐ Invite my Address Book contacts to Spock





Find Friends

**Gmail**

Username:

Password:

[Forgot your password?](#)

**Sign In**

We will not store your login. It will only be temporarily used to access your Gmail address book.

Switch Accounts:  
[gmail](#) | [yahoo!](#) | [hotmail](#) | [aol](#) | [outlook](#) | [other](#)

**Close**

## Brightkite & Twitter

Twitter username

*Your Twitter username.*



Twitter password

*The password you use to log into Twitter. We need this so we can post updates on your behalf.*

Also post to Twitter

- ☐ When I post a note.
- ☐ When I post a photo.
- ☐ When I check in at a place, post my checkin to Twitter.
- ☐ When I check in at a place, set my Twitter location accordingly (NEW).

*Choose which Brightkite actions you want to post to your Twitter account. Keep in mind that this might easily annoy your followers, so choose wisely.*

Save



What's on your mind ?



**"Giving your email account password to a social network site so they can look up your friends is the same thing as going to dinner and giving your ATM card and PIN code to the waiter when it's time to pay."**

- oauth.net





we need an easy,  
user-friendly standard  
for third party api security



# OAuth usage





OAUTH

[About](#) [Advisories](#) [Documentation](#) [Code](#) [Blog](#) [Community](#)

An **open protocol** to allow **secure API authorization** in a **simple** and **standard** method from desktop and web applications.

[Read the specification »](#)

For Consumer developers...

If you're building...

- desktop applications
- dashboard widgets or gadgets
- Javascript or browser-based apps
- webpage widgets

OAuth is a simple way to publish and interact with protected data. It's also a safer and more secure way for people to give you access. We've kept it simple to save you time.

For Service Provider developers...

If you're supporting...

- web applications
- server-side APIs
- mashups

If you're storing protected data on your users' behalf, they shouldn't be spreading their passwords around the web to get access to it. Use OAuth to give your users access to their data while protecting their account credentials.

[Get started...](#)

Learn more about the emerging [OAuth 2.0 work](#).



OAuth puts the user  
back in control

You choose who you share  
your data with

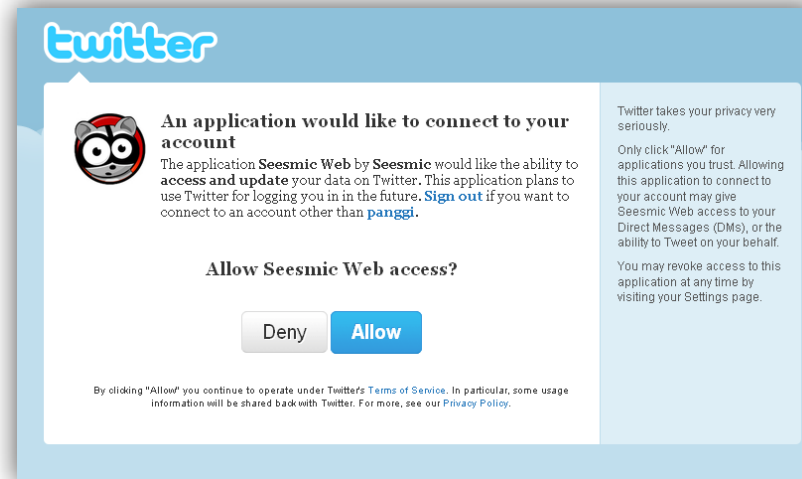
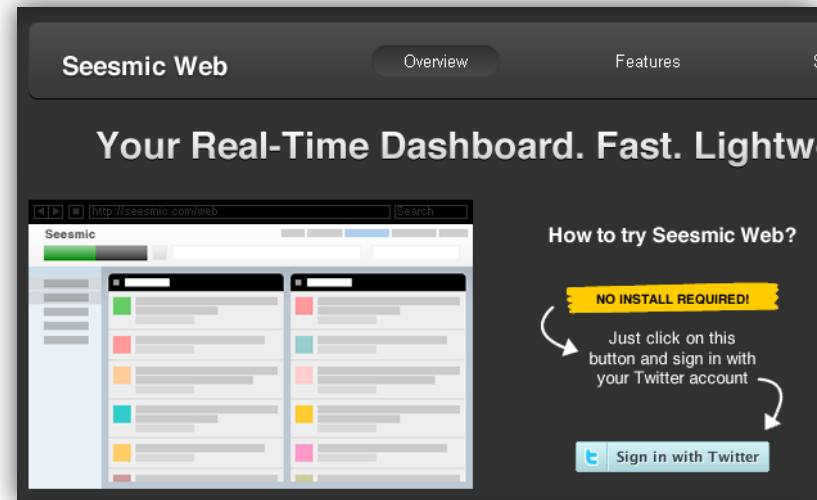


The screenshot shows the Twitter interface for a user named 'panggi'. The 'Connections' tab is selected, displaying a list of applications that have been granted access to the user's account. Each entry includes the application's icon, name, developer, description, and a 'Revoke Access' link.

Application	Developer	Description	Access Granted
Dabr	David Carrington	A mobile front end to Twitter optimised to bring full functionality to a wide selection of handsets.	approved on 2:38 PM Oct 5th · read and write access
Apigee's API Console	Apigee	Explore the structure of the Twitter API, experiment with the endpoint, and review the request and response messages from inside your browser.	approved on 11:15 AM Sep 16th · read and write access
Seesmic Desktop	Seesmic	The desktop client for your social networking	approved on 11:50 AM Sep 12th · read and write access
The Engineering Blog	Blog	An @Anywhere application	approved on 8:21 PM Aug 23rd · read and write access
Koprol		Location-based social network	approved on 5:04 PM Aug 5th · read and write access

# OAuth is secure

No need to give  
Username and  
Password



Default Access type:

☒ Read & Write ☐ Read-only

What type of access does your application need? Note: @Anywhere applications require read & write access.

# Big Name Adoption

Google

OpenSocial

MySpace

SmugMug

Yahoo!

Netflix

twitter

GetSatisfaction *and more...*

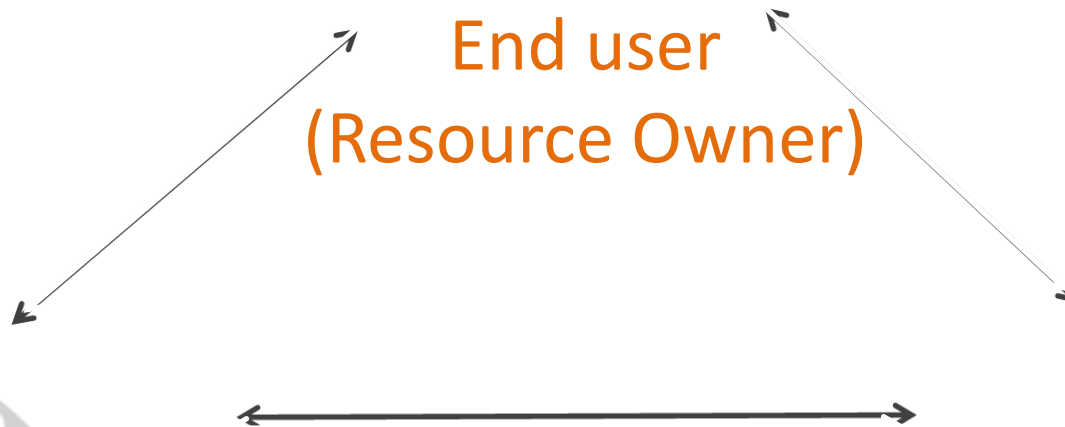


# OAuth

Love triangle



End user  
(Resource Owner)



Consumer



# OAuth

Protected resources  
are exposed by service providers  
and used by consumer applications  
on behalf of users



# OAuth

My Twitter Status

Is exposed by Twitter

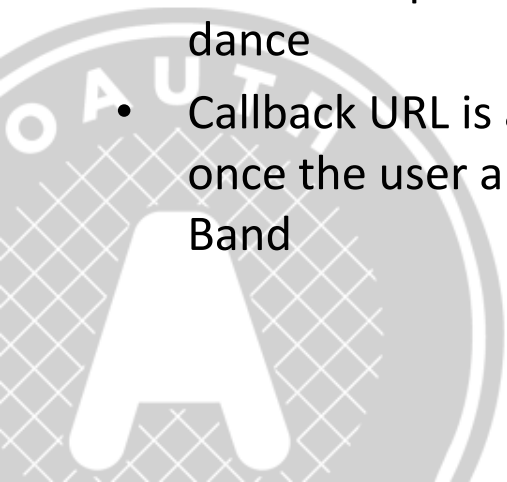
And used by Seesmic

On my behalf



# OAuth Terminology

- Provider is the application that exposes the secure API and user's identity
- Consumer is the application that is written against the Provider's API, intended for Provider's users.
- Users or resource owners are registered users of the Provider
- Consumer Key is an identifier for the consumer
- Consumer Secret is a shared-secret between the provider and the consumer
- Signature Methods are encryption methods used by OAuth communication. Methods suggested are PLAINTEXT, RSA-SHA1 and HMAC-SHA1
- OAuth Endpoints are endpoints exposed by the provider to facilitate OAuth dance
- Callback URL is an endpoint at the Consumer that is invoked by the Provider once the user authorizes the Consumer. If none, the value is oob, or Out-of-Band



# Tokens

- Request Token
  - Short lived identifiers which start the handshake
  - Must be converted to Access Token in order to gain access to a user's resources
- Access Token
  - Long lived identifiers that are tied to the user's identity
  - Are used to access a user's resources (data) at the Provider on behalf of the user



# Endpoints

- Get request token
- Authorize token
- Get access token



# Get Request Token

- The endpoint provides consumers to get an unauthorized request token by providing their consumer key and other parameters as a signed request
- The credentials can be passed via HTTP Header, POST body or GET QueryString
- The request includes an oauth\_signature which is calculated by following the steps defined in the spec. Use libraries instead of writing your own signing implementations.
- The response has an unauthorized request token as well as a token secret, and a flag indicating if the callback was accepted.



# Authorize Token

- The step authorizes an unauthorized request token retrieved via previous request.
- The endpoint takes the unauthorized request token – or the user can enter one manually if supported.
- The Authorize Token endpoint then redirects the user to the Provider's login page
- The user logs in, and is asked to authorize the consumer (and hence the request token)
- Once the user authenticates, and authorizes access to the consumer, the provider calls the callback URL provided earlier with a verifier code. This verifier code, along with other credentials is used to get an Access Token.



# Get Access Token

- At this step, the now authorized request token is exchanged for an access token
- The access token acts as a user's credential for any further transactions
- The endpoint takes the request token and the verifier code returned via the callback, or manually if callback is not supported. The request is signed with consumer secret and the request token's secret.
- The Provider returns an access token and a token secret.
- The token secret is used to sign the requests along with the consumer secret.

# Access User's Resources

- Now that the consumer has the access token, the user's resources can be requested via signed requests to the provider.
- The user should be able to unauthorize the consumer by revoking the access token.
- The access token has a time to live which is typically longer than the request token



# Useful Resources



- <http://tools.ietf.org/html/rfc5849>
- <http://oauth.net/code/>
- <http://hueniverse.com/oauth/>
- <http://code.google.com/p/oauth/>
- <http://opensecuritylab.org/tag/oauth>

