Twitter @anywhere

by Aaron Saray
for web414.com

aaronsaray.com
@aaronsaray

Who am I?

- Organizer of Milwaukee PHP Users Group
- Author (aaronsaray.com/book)

- Programmer since I was 8

- ECMA is my middle name

- PHP/JS 1 year shy of a decade

Why should you trust me?

- I do javascript
- I like twitter (@aaronsaray)
- I used AJAX before it had a name
  - JEMLinx.com
- I just finished implementing what I'm going to teach you

What is @anywhere

- Allows users to engage with Twitter on your website safely
- "Aggressively simple"
- It's like Facebook Connect
  - Only with more FailWhale
- Demoed at Chirp

That is just TOO fluffy

- Linkify Users
- Hovercards
- follow buttons
- tweetbox
- authentication callbacks

Steps to Get Started

- Create an application on dev.twitter.com
- Add javascript tag with API key
- decide/implement features you want
- {magic}
- profit

Step 1: Create Application

- visit http://dev.twitter.com/apps after authentication
- Click on Register a new application

Step 2: Fill in the fields

twitter developers

API Status | Documentation | Discussions | Your apps | Sign out

Search

Register an Application
Create your own Twitter app

Tell us about your application.

View your applications

Application Name: MKEPUG.org

Description: Link your tweet up with us.

Application Website: http://mkepug.org

Where's your application's home page, where users can go to download or use it?

Organization: Milwaukee PHP Users Group

Application Type: ○ Client ● Browser

Does your application run in a Web Browser or a Desktop Client?

Browser uses a Callback URL to return to your App after successful authentication.

Client prompts your user to return to your application after approving access.

Callback URL: http://mkepug.org

Where should we return to after successfully authenticating?

You can override this at any time by sending an *oauth_callback* while obtaining a request_token.

Default Access type: ● Read & Write ○ Read-only

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

Application Icon: You can upload this later!

C:\Documents and Settings\Aarc Browse...

Maximum size of 700k. JPG, GIF, PNG.

twomey War

twomey War

Can't read this?
Get two new words
Hear a set of words
Powered by reCAPTCHA
Help

Step 2 - continued

- Application Name
  - What will show up as 'from'
  - twitroid, tweetdeck, etc.
- Application Website
  - The clickthrough on the name
- Callback URL
  - Mainly useful with oAuth - just put your main URL
- set read&write
  - read can only read the stream - we want to update
- click register!

Step 3: Done

MKEPUG.org Settings
By Milwaukee PHP Users Group

Application details

Link your tweet up with us
Created by Milwaukee PUG

Edit Application Settings

Reset Consumer Key/Secret

My Access Token

View your applications

@Anywhere Settings

@Anywhere is easy to deploy. You only need an API key and registered callback URL.

API key

Registered Callback URL

http://mkepug.org
The @Anywhere callback URL's domain & subdomain must match the location of @Anywhere integrations on your site.

OAuth 1.0a Settings

Step 3 - continued

- Grab your API Key
- Put it in this code replacing 'YOUR_API_KEY'

<script src="http://platform.twitter.com/anywhere.js?id=YOUR_API_KEY&v=1 type="text/javascript"></script>

- note: obviously include this JS before you call the twitter object
- hint: optimization wise, its best to include your JS at the bottom of the page - so do that with this too!
- other note: see v=1 for version 1.
  - there also is a v=chirp_preview

OK - Important Note

- The API documentation is here:
  - http://dev.twitter.com/anywhere/begin
- A shortcut reference is here:
  - http://platform.twitter.com/js-api.html
- **The most important thing I can tell you about @anywhere is next slide**

It is NOT DONE

- 4 hours later I checked a forum
- don't be dumb
- use google groups
- I suck.

The twttr object

- The twttr object is how you refer to @anywhere
- Can be called multiple times
- Accepts a callback

```
twttr.anywhere(onLoadAnywhere);
function onLoadAnywhere(T) {}
```

or

```
twtr.anywhere(function(T){});
```

The twttr object uses selectors

- It uses a selector to be more specific during invocation
- it uses Sizzle:
  - http://wiki.github.com/jeresig/sizzle/
- If you use jQuery or any library, you know how to do this:

```
jquery: $("#something");
twttr: T("#something");
```

Auto Linkify Usernames

- Just a click through: boring

```
twttr.anywhere(function(T){
    T.linkifyUsers();
});
```

- want to linkify users only in a profile box with id 'profilebox'?

```
twttr.anywhere(function(T){
    T("#profilebox").linkifyUsers();
});
```

Hovercards

- awesomesauce compared to linkify

```
twttr.anywhere(function(T){
    T.hovercards();
});
```

Follow Buttons

- Follow button allows the person to follow you directly from the site
- This is also done with hovercards
- updates if you are following that person (you can't refollow)

```
<span id="follow-twitterapi"></span>
<script type="text/javascript">

 twttr.anywhere(function (T) {
   T('#follow-twitterapi').followButton("twitterapi");
 });

</script>
```

Tweetbox: My Favorite

- Loads an iframe on your site with tweetbox
- kinda ugly right now
- requires extensive javascript to edit portions of it

```
<div id="tbox"></div>
<script type="text/javascript">
 twttr.anywhere(function (T) {
   T("#tbox").tweetBox({
     height: 100,
     width: 400,
     defaultContent: "I Really like @web414.  @aaronsaray rocks too"
   });
 });
</script>
```

Tweetbox

Default
What's happening?   140

[text input box]

                                        [Tweet]

<-- ehhh...

Tweetbox Options
- counter: boolean - whether to show the counter or not
- height/width: size in px of the tweetbox
- label: string label of the box
  - cool tip - it accepts HTML ;)
- defaultContent: plain text prepopulation
- onTweet: function to execute after success
  - receives plain text and html of the tweet
  - also useful for giving a friendly 'good job' success after tweet
- data: object - should contain information about the tweet...

Twitter Connection: authComplete / signout

- When the user connects using a connect button
- you can use a connect button to offer advanced API functionality ahead of time
  - don't need it for tweetbox - need it for user data
- two callbacks:
  - authComplete
  - signOut

Twitter connection example:

```
twttr.anywhere(function(T) {
    T("#connect").connectButton({
        authComplete: function() {
            $("#connect").hide();
            $("#disconnect").html('<a href="#" id="signoutlink"">Sign Out</a>');
        },
        signOut: function() {
            $("#connect").show();
        }});

    if (T.isConnected()) {
        $("#connect").hide();
        $("#disconnect").html('<a href="#" id="signoutlink">Sign Out</a>');
    }
});

$("#signoutlink").live("click", function () { twitterSignout(); });

function twitterSignout() {
    twttr.anywhere.signOut();
    $("#disconnect").hide(); $("#connect").show(); $("#connectedUser").html(");
}
```

and a tad bit more...

```
<div id="connect"></div>
<div id="disconnect"></div>
```

- Shows a connect button. When connected, hides the connect button and shows a signout link.
- Next time the page is loaded, isConnected() does the same sorta thing.
- The logout button disconnects and shows the connect box.

Interacting with the current user

- This is based off of the object passed to your twttr function callback
- I like to use T as the object
- You might find that this is useful:http://platform.twitter. com/js-api.html
  - Hint: It's not.
  - Better hint: check out my proof of concept for much more details:http://aaronsaray.com/blog/2010/05/13/twitter- anywhere-proof-of-concepts/
- They must be connected to interact with them
- The current visitor must be connected to view other's information
  - besides hovercards()

Finding user information

```
twttr.anywhere(function(T) {
    if (!T.isConnected()) {
        alert('you can only use this if you are connected');
        return false;
    }
    T.User.find(username, function(user) {
        alert("user has " + user.attributes.friends_count + " friends");
        alert("user is following " + user.attributes.followers_count + " tweeple");
        alert("you are following them? " + user.attributes.following);
        alert(user.attributes.name + " last said: " + user.attributes.status.text);
    });
});
```

Last time I checked these were all the things that would work.

Things that the documentation says that aren't correct or finished yet

- screenName = currentUser.data('screen_name');
  - data object is not working yet. use: user.attributes. screen_name instead
- It's easy to use for anyone
  - not exactly. On load, users can't click tweet. Needs to be focused. Try this code:

```
setTimeout(function() {
    $("#tweetbox").focus();
    $(".twitter-anywhere-tweet-box").contents().find("#tweet-box").focus();
}, 1000);
```

Now they can click the button

What I didn't cover

- Using twitter to fully authenticate a user using @anywhere
  - I think this is bad.

One Quick Example

http://markforgov.com/contact

Questions?

Otherwise, this is the end...