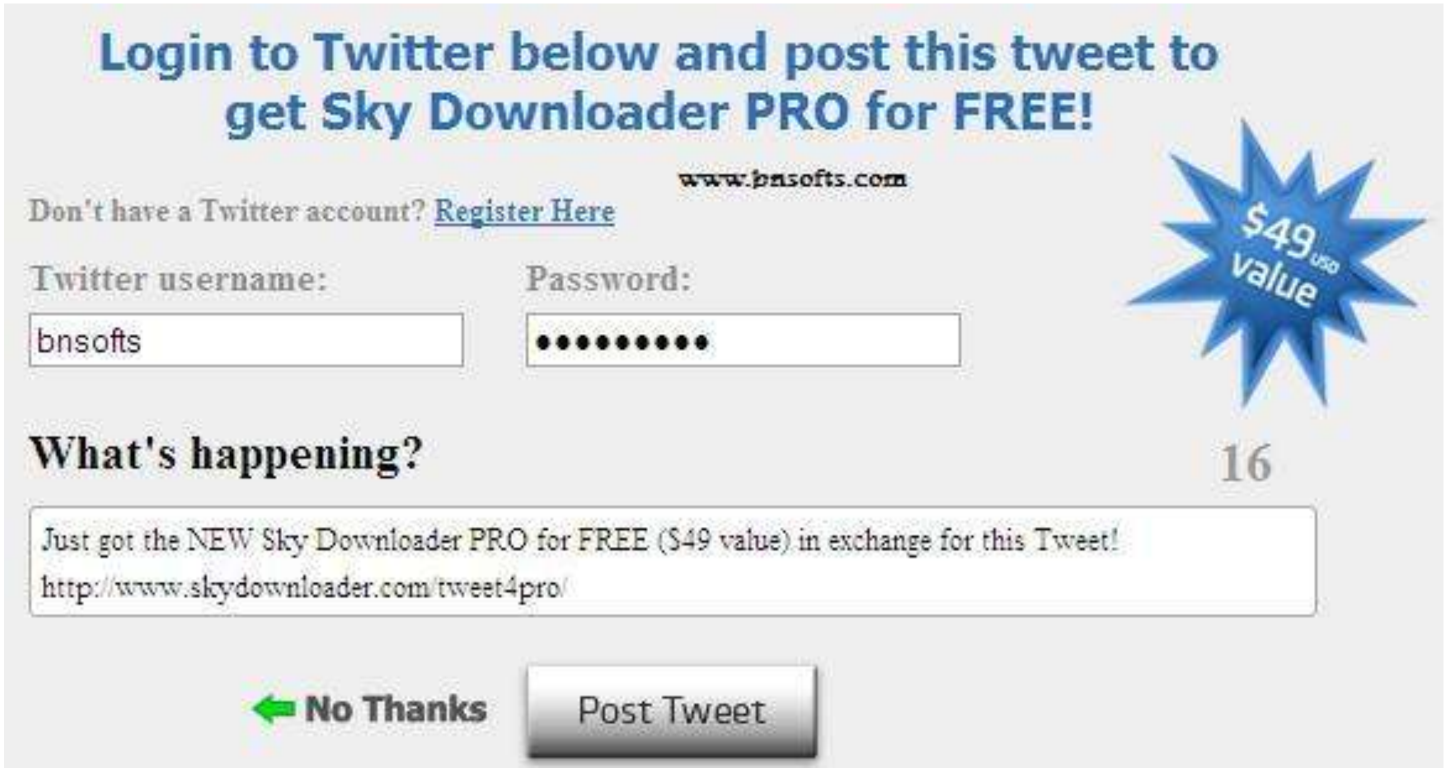
An Introduction to OAuth 2

Aaron Parecki • @aaronpk
OSCON • Portland, Oregon • July 2012

A Brief History


Login to Twitter below and post this tweet to get Sky Downloader PRO for FREE!

www.bnsofts.com

Don't have a Twitter account? Register Here

Twitter username: bnsofts

Password: ••••••••••

What's happening? 16

Just got the NEW Sky Downloader PRO for FREE ($49 value) in exchange for this Tweet! http://www.skydownloader.com/tweet4pro/

Before OAuth

aka the Dark Ages

If a third party wanted access to an account, you'd give them your password.

Several Problems and Limitations

- Apps store the user’s password
- Apps get complete access to a user’s account
- Users can’t revoke access to an app except by changing their password
- Compromised apps expose the user’s password

Before OAuth 1.0

- Services recognized the problems with password authentication
- Many services implemented things similar to OAuth 1.0
- Each implementation was slightly different, certainly not compatible with each other

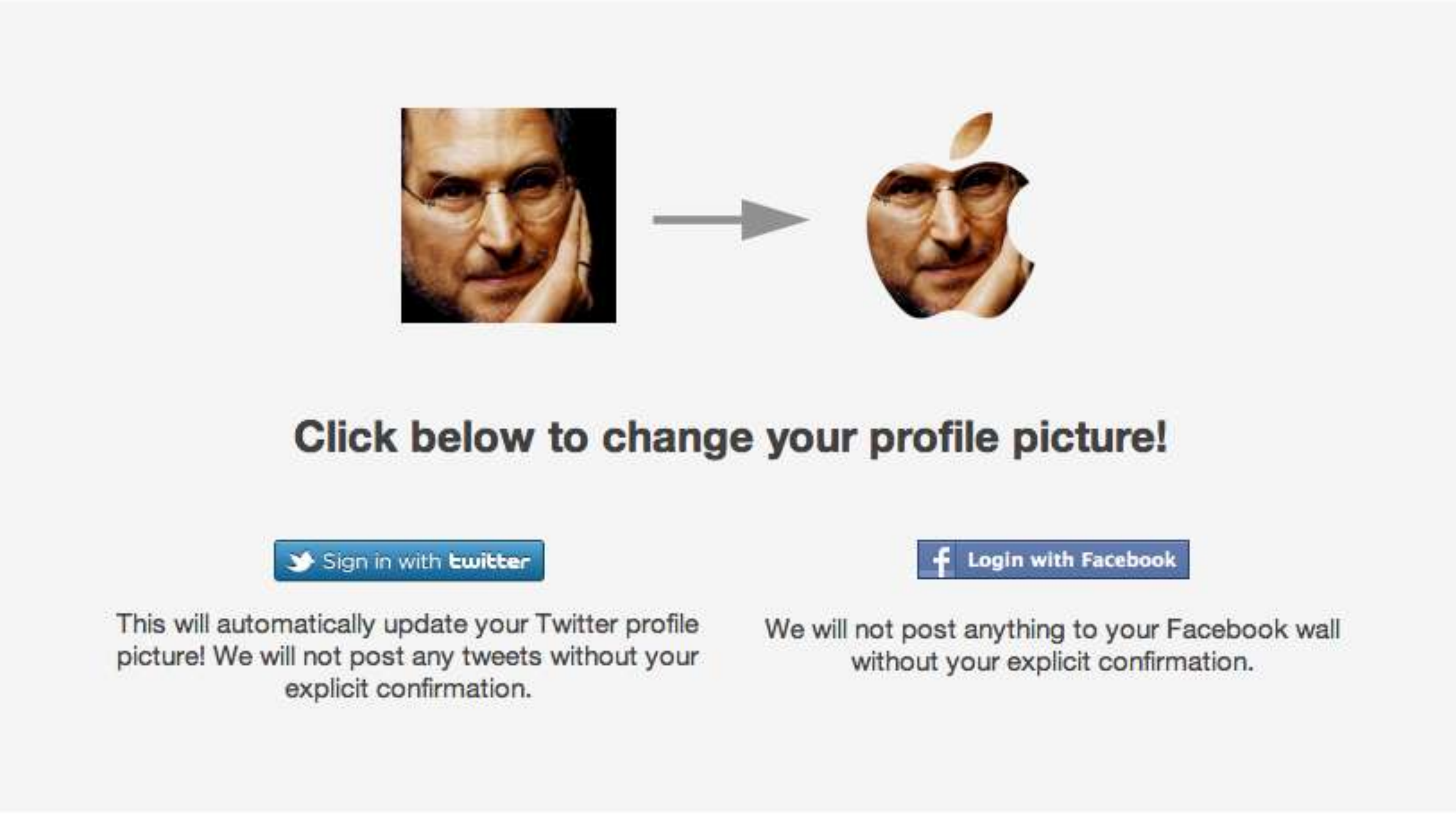
Before OAuth 1.0

- Flickr: "FlickrAuth" frobs and tokens
- Google: "AuthSub"
- Facebook: requests signed with MD5 hashes
- Yahoo: BBAuth ("Browser-Based Auth")

"We want something like Flickr Auth / Google AuthSub / Yahoo! BBAuth, but published as an open standard, with common server and client libraries."

Blaine Cook, April 5th, 2007

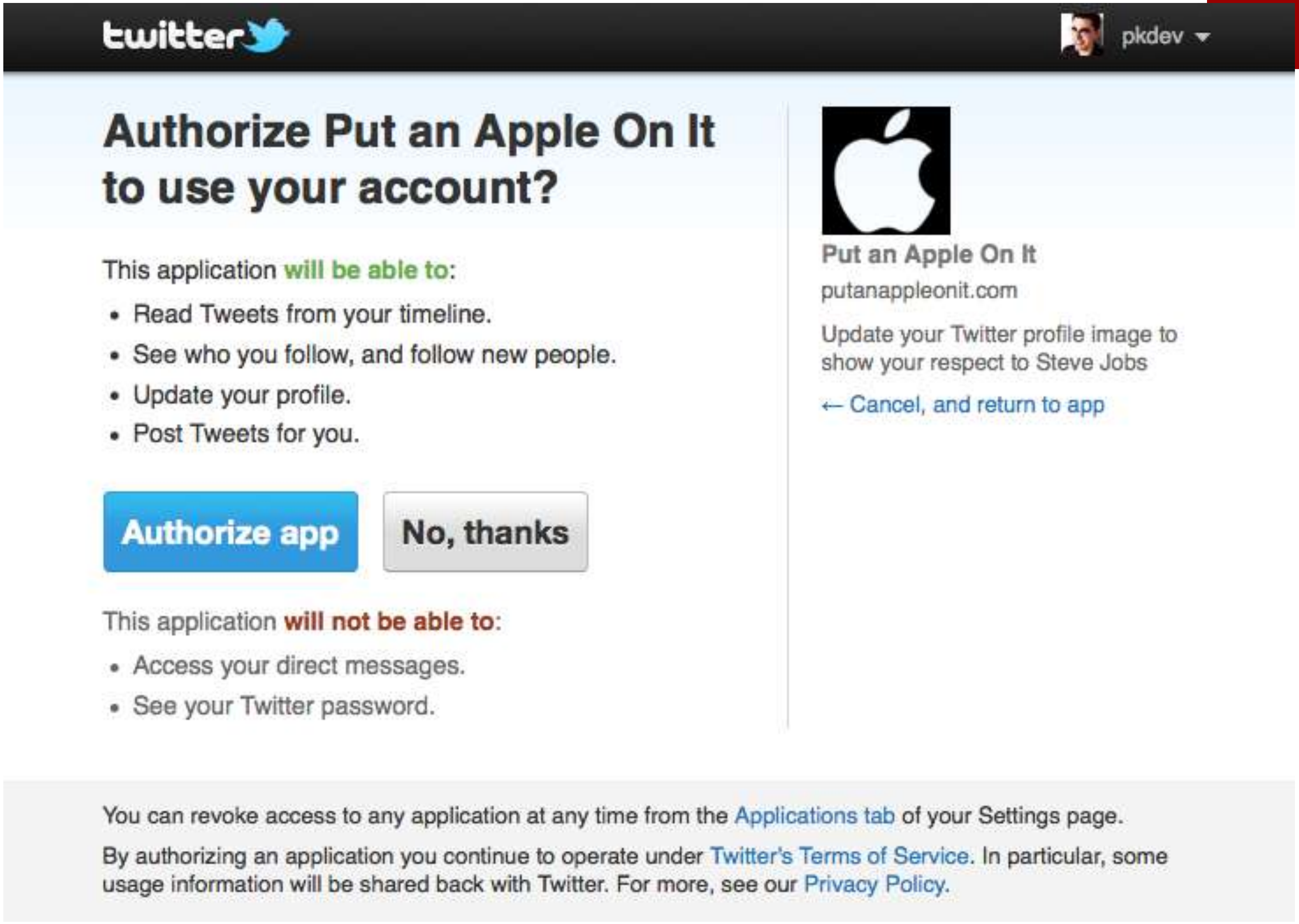
OAuth 1.0



Click below to change your profile picture!

This will automatically update your Twitter profile picture! We will not post any tweets without your explicit confirmation.

We will not post anything to your Facebook wall without your explicit confirmation.



Authorize Put an Apple On It to use your account?

This application **will be able to**:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

This application **will not be able to**:

- Access your direct messages.
- See your Twitter password.

Put an Apple On It

putanappleonit.com

Update your Twitter profile image to show your respect to Steve Jobs

← Cancel, and return to app

You can revoke access to any application at any time from the Applications tab of your Settings page.

By authorizing an application you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Your Twitter profile picture was updated!

RIP Steve Jobs. Show your love by changing your profile picture: http://putanappleonit.com

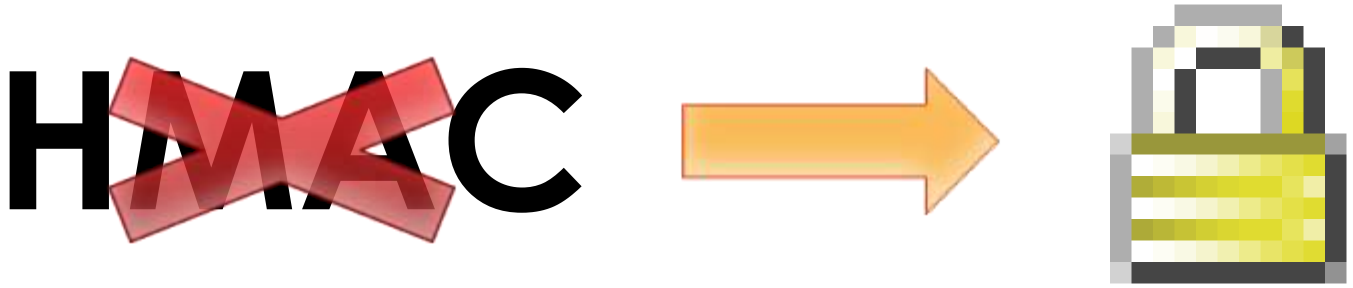
Tweet This!

OAuth 1.0 Signatures

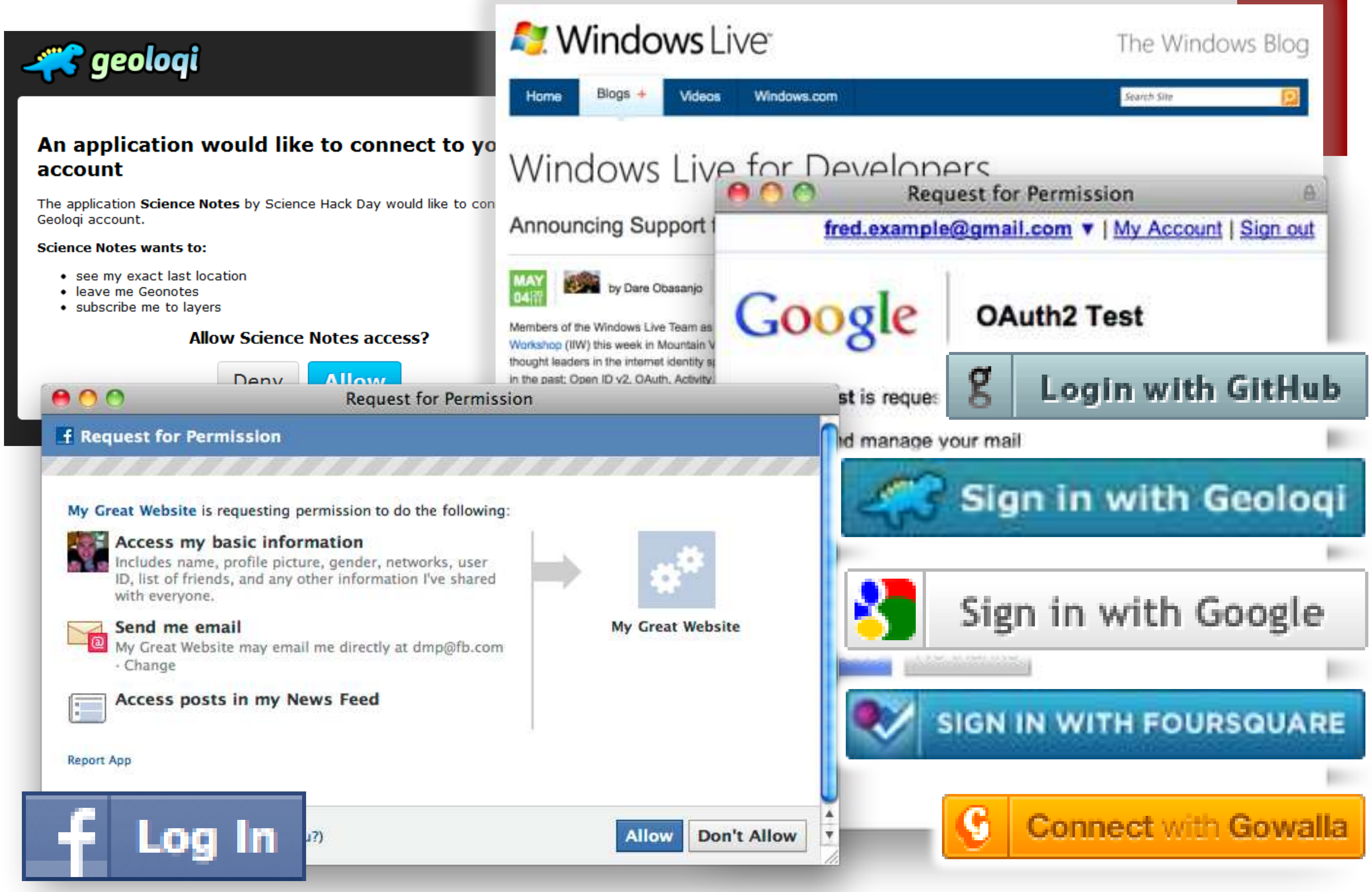
The signature base string is often the most difficult part of OAuth for newcomers to construct. The signature base string is composed of the HTTP method being used, followed by an ampersand ("&") and then the URL-encoded base URL being accessed, complete with path (but not query parameters), followed by an ampersand ("&"). Then, you take all query parameters and POST body parameters (when the POST body is of the URL-encoded type, otherwise the POST body is ignored), including the OAuth parameters necessary for negotiation with the request at hand, and sort them in lexicographical order by fi
then parameter value (for duplica
while ensuring that both the key a
parameter are URL encoded in iso
the equals ("=") sign to mark the ke
use the URL-encoded form of "%3D
joined by the URL-escaped amper

```
oauth_nonce="QP70eNmVz8jvdPevU3oJD2AfF7R7o
dC2XJcn4XlZJqk", oauth_callback="http%3A%2F%2
Flocalhost%3A3005%2Fthe_dance%2Fprocess_callb
ack%3Fservice_provider_id%3D11", oauth_signatur
e_method="HMAC-
SHA1", oauth_timestamp="1272323042", oauth_cons
umer_key="GDdmIQH6jhtmLUypg82g", oauth_signa
ture="8wUi7m5HFQy76nowoCThusfgB%2BQ%3D", oa
uth_version="1.0"
```

OAuth 2:
signatures replaced by https



Some Current Implementers


geoloqi

An application would like to connect to yo
account

The application **Science Notes** by Science Hack Day would like to con
Geoloqi account.

Science Notes wants to:

- see my exact last location
- leave me Geonotes
- subscribe me to layers

Allow Science Notes access?

Windows Live — The Windows Blog

Windows Live for Developers

Announcing Support f

by Dare Obasanjo

Members of the Windows Live Team as
Workshop (IIW) this week in Mountain V
thought leaders in the internet identity s
in the past: Open ID v2, OAuth, Activity

Request for Permission

fred.example@gmail.com ▼ | My Account | Sign out

Google OAuth2 Test

st is reques

d manage your mail

Request for Permission

Request for Permission

My Great Website is requesting permission to do the following:

Access my basic information
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

Send me email
My Great Website may email me directly at dmp@fb.com · Change

Access posts in my News Feed

My Great Website

Report App

u?)

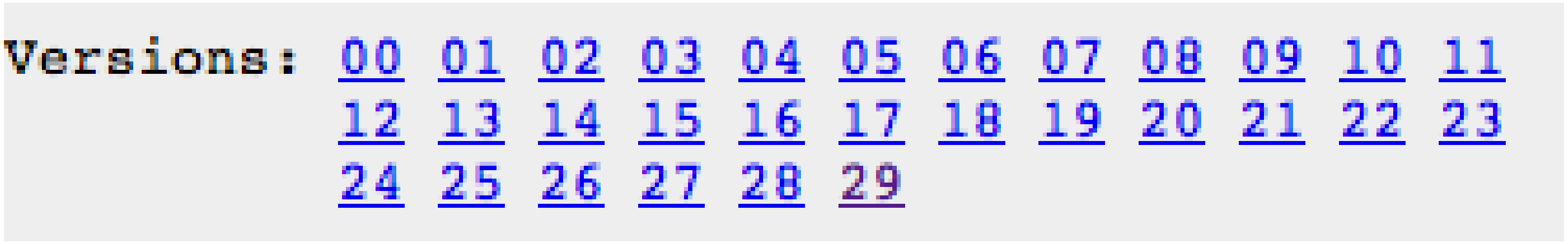
The OAuth 2 Spec

http://oauth.net/2/

Table of Contents

OAuth 2?!

There are 29 versions!



```
Versions: 00 01 02 03 04 05 06 07 08 09 10 11
          12 13 14 15 16 17 18 19 20 21 22 23
          24 25 26 27 28 29
```

Currently Implemented Drafts

Provider	Draft	Reference
Foursquare	-10	http://aaron.pk/2YS
Google	-10	http://code.google.com/apis/accounts/docs/OAuth2.html
Facebook	-10 (ish)	https://developers.facebook.com/docs/authentication/oauth2_updates/
Windows Live	-10	http://aaron.pk/2YV
Salesforce	-10	http://aaron.pk/2YW
Github	-07	http://develop.github.com/p/oauth.html
Geoloqi	-10	http://developers.geoloqi.com/api

So how does it work?

Definitions

- **Resource Owner:** The User
- **Resource Server:** The API
- **Authorization Server:** Often the same as the API server
- **Client:** The Third-Party Application

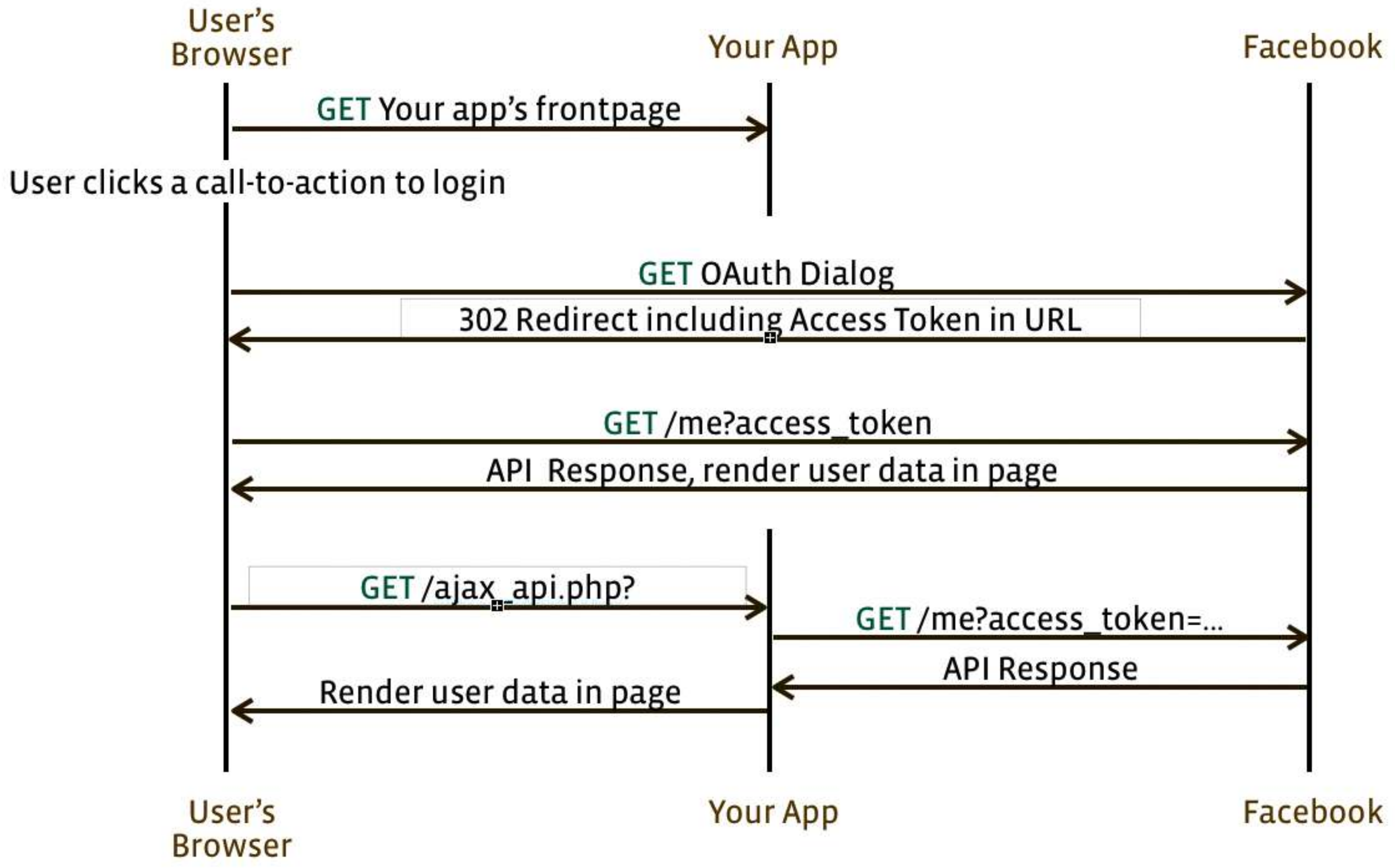
Use Cases

- Web-server apps
- Browser-based apps
- Username/password access
- Application access
- Mobile apps

Use Cases – Grant Types

- Web-server apps – `authorization_code`
- Browser-based apps – `implicit`
- Username/password access – `password`
- Application access – `client_credentials`
- Mobile apps – `implicit`

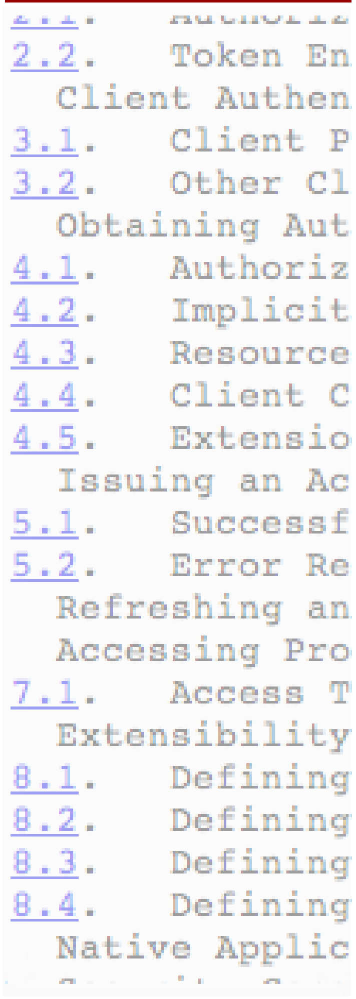
Facebook's OAuth Flow



Source: https://developers.facebook.com/docs/authentication/

Web Server Apps

Authorization Code Grant

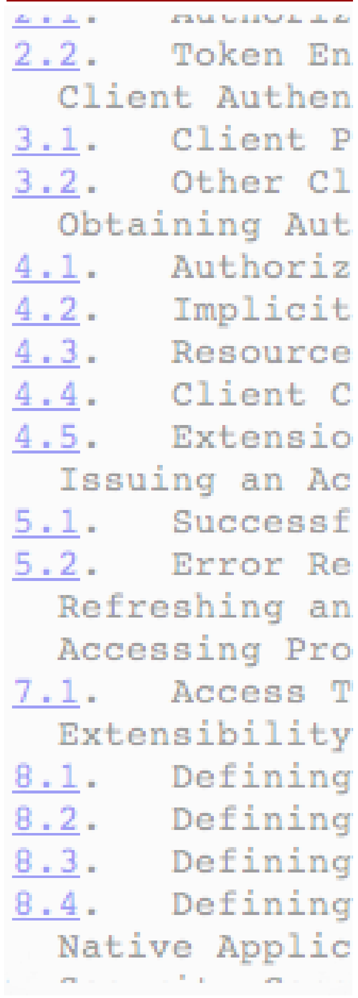


2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

Create a "Log In" link

Link to:

```
https://facebook.com/dialog/oauth?response_
type=code&client_id=YOUR_CLIENT_ID&redirect
_uri=REDIRECT_URI&scope=email
```

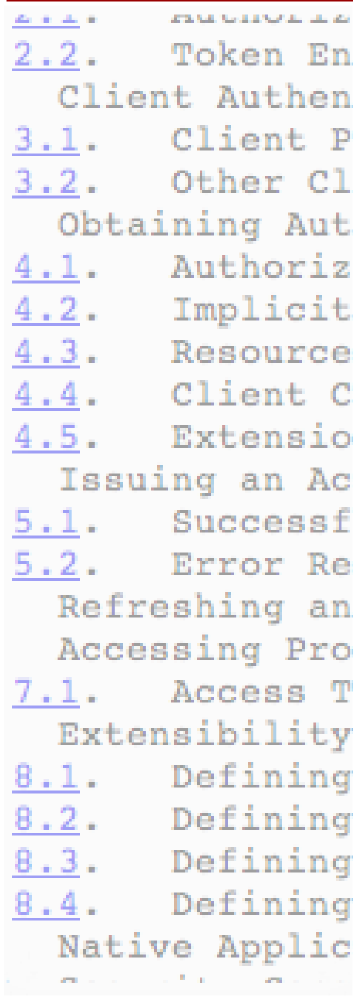

2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

Create a “Log In” link

Link to:

https://facebook.com/dialog/oauth?**response_type=code**&client_id=YOUR_CLIENT_ID&redirect_uri=REDIRECT_URI&scope=email

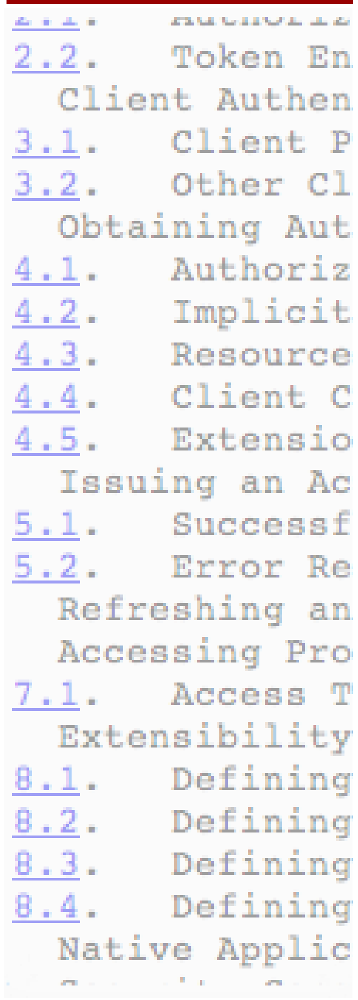
Login with Facebook



Create a “Log In” link

Link to:

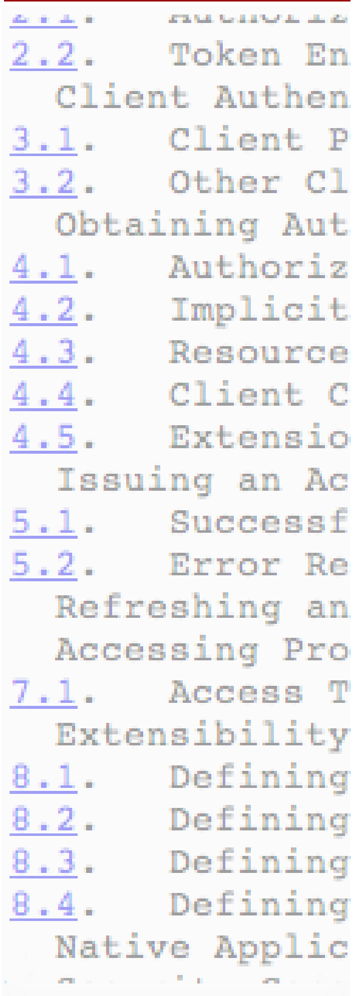
https://facebook.com/dialog/oauth?response_type=code&**client_id=YOUR_CLIENT_ID**&redirect_uri=REDIRECT_URI&scope=email


2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

Create a “Log In” link

Link to:

https://facebook.com/dialog/oauth?response_type=code&client_id=YOUR_CLIENT_ID&**redirect_uri=REDIRECT_URI**&scope=email



Create a "Log In" link

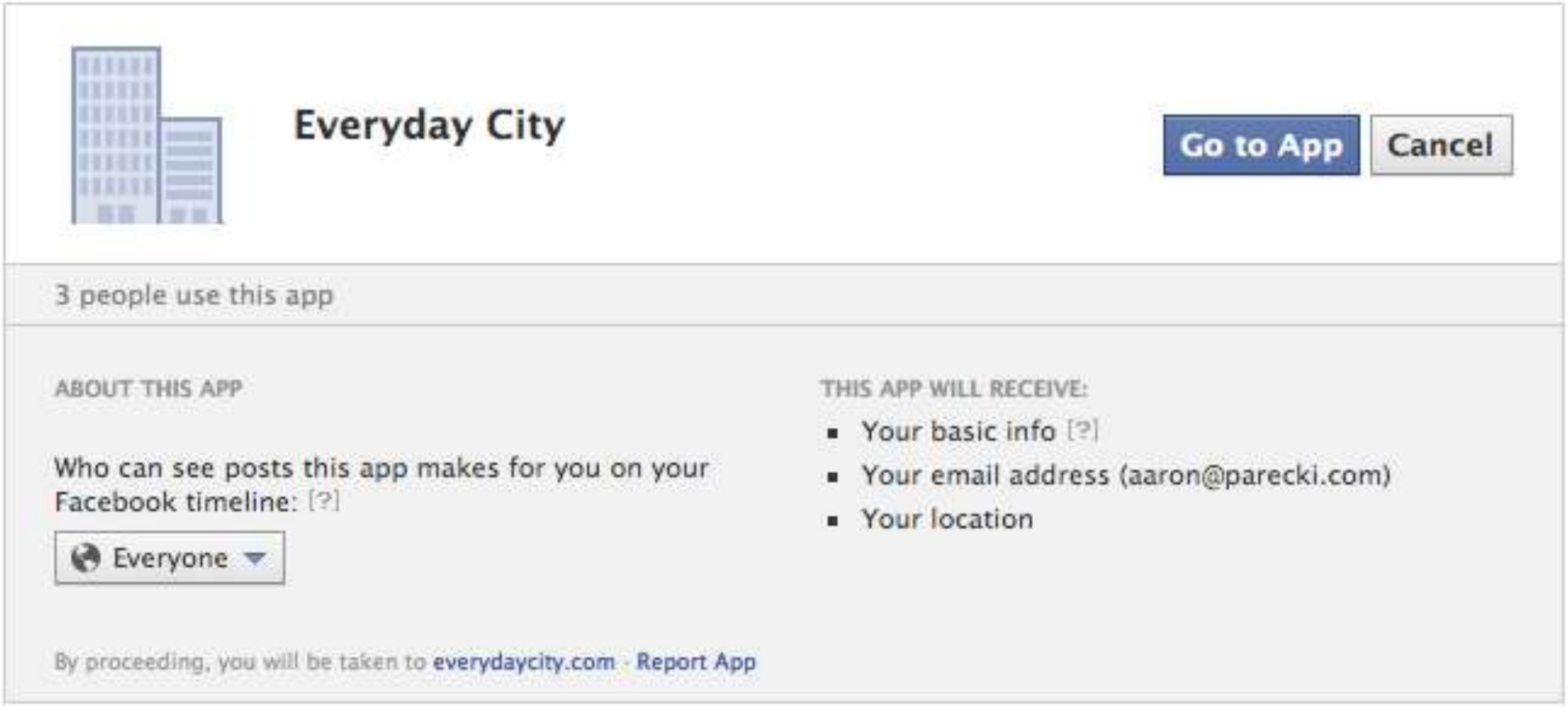
Link to:

https://facebook.com/dialog/oauth?response_type=code&client_id=YOUR_CLIENT_ID&redirect_uri=REDIRECT_URI&**scope=email**

Login with Facebook

User visits the authorization page

```
https://facebook.com/dialog/oauth?response_
type=code&client_id=28653682475872&redirect
_uri=everydaycity.com&scope=email
```


Everyday City

Go to App | Cancel

3 people use this app

ABOUT THIS APP

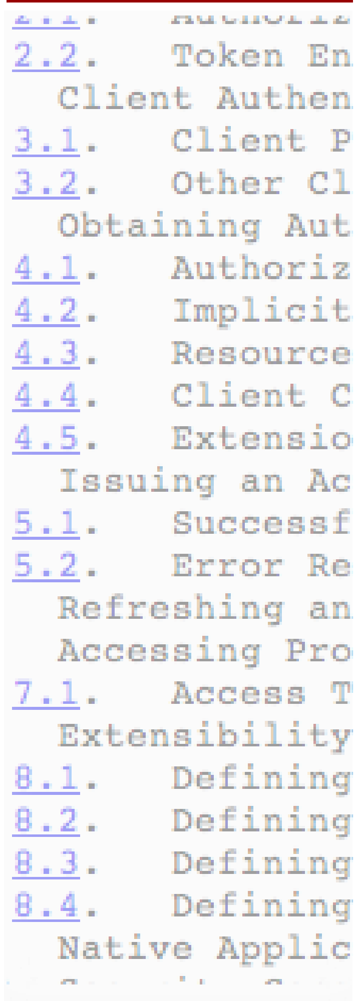
Who can see posts this app makes for you on your Facebook timeline: [?]

Everyone

THIS APP WILL RECEIVE:

- Your basic info [?]
- Your email address (aaron@parecki.com)
- Your location

By proceeding, you will be taken to everydaycity.com · Report App

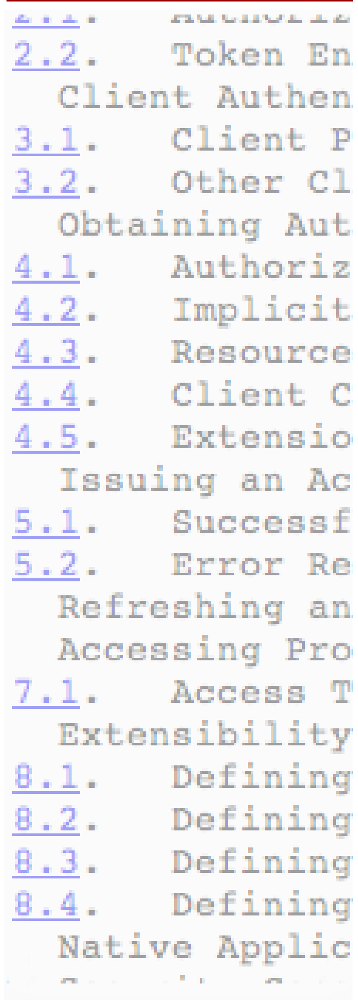

2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

On success, user is redirected back to your site with auth code

```
https://example.com/auth?code=AUTH_CODE_HERE
```

On error, user is redirected back to your site with error code

```
https://example.com/auth?error=access_denied
```



2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

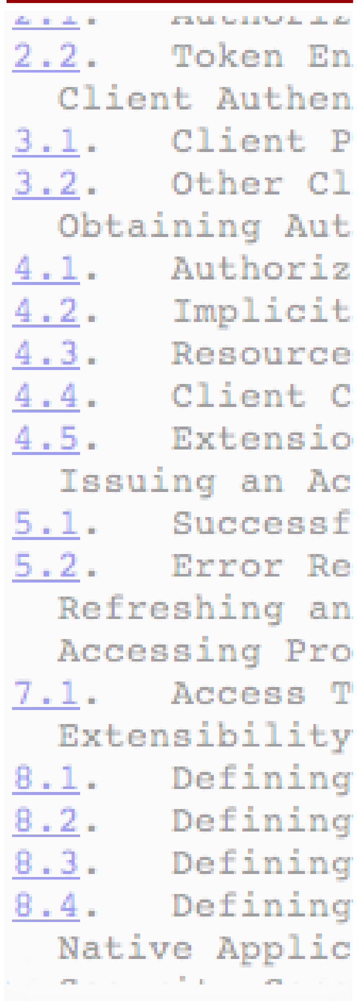
Server exchanges auth code for an access token

Your server makes the following request

```
POST
https://graph.facebook.com/oauth/access_to
ken

Post Body:
grant_type=authorization_code
&code=CODE_FROM_QUERY_STRING
&redirect_uri=REDIRECT_URI
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

Server exchanges auth code for an access token



Your server gets a response like the following

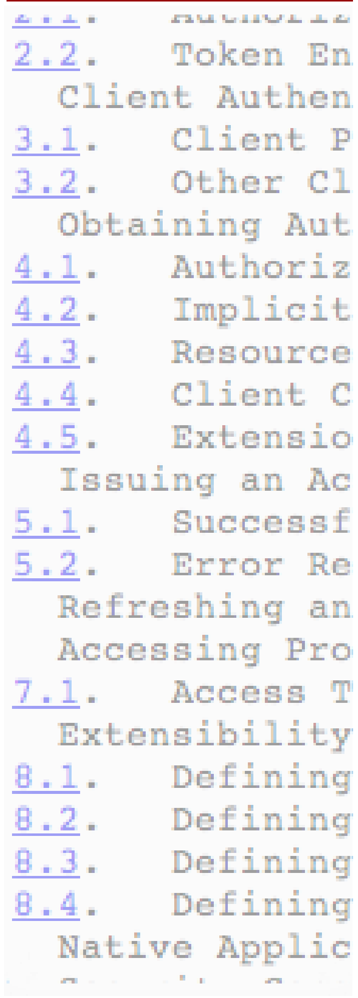
```
{
  "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"e1qoXg7Ik2RRua48lXIV"
}
```

or if there was an error

```
{
  "error":"invalid_request"
}
```

Browser-Based Apps

Implicit Grant



Create a "Log In" link

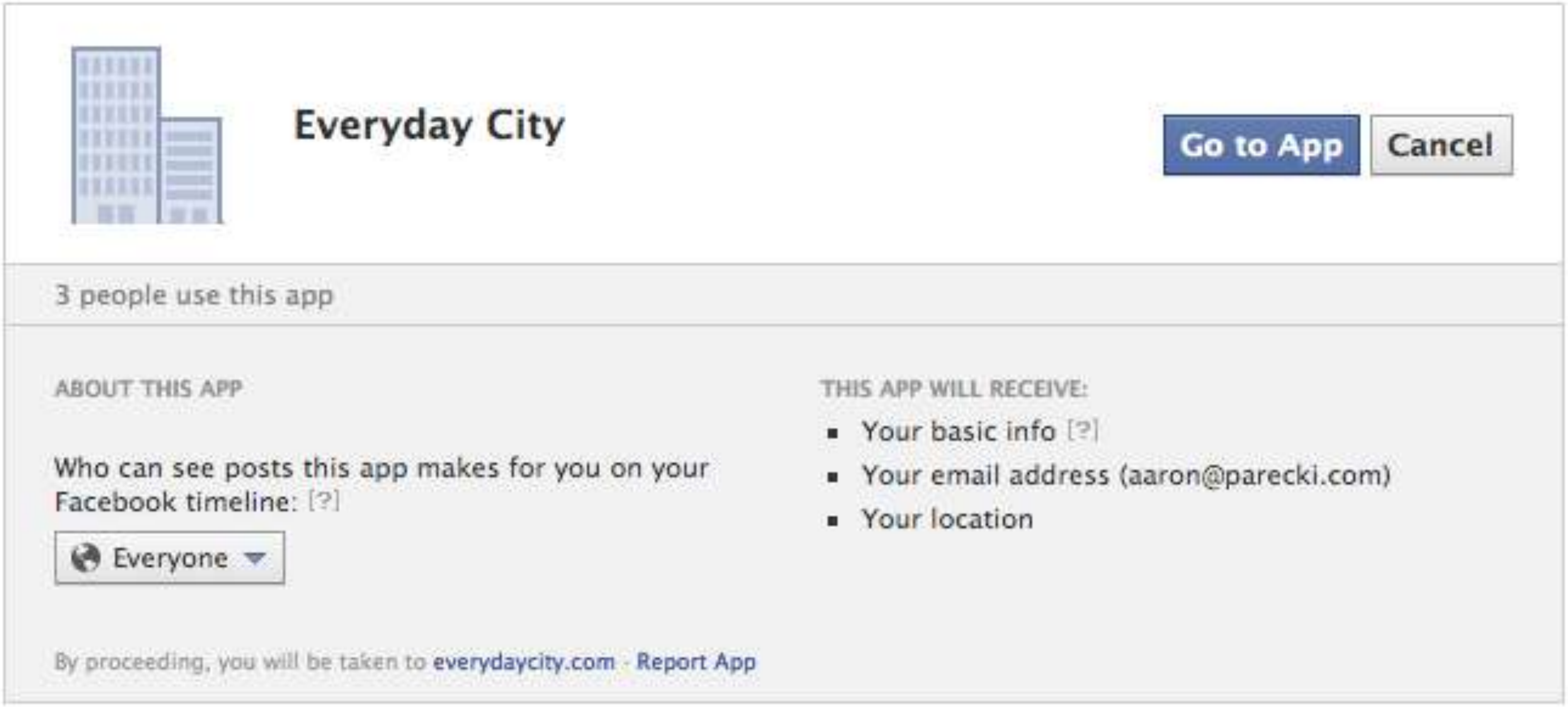
Link to:

```
https://facebook.com/dialog/oauth?**response_type=token**&client_id=CLIENT_ID
&redirect_uri=REDIRECT_URI&scope=email
```

Login with Facebook

User visits the authorization page

```
https://facebook.com/dialog/oauth?response_
type=token&client_id=2865368247587&redirect
_uri=everydaycity.com&scope=email
```



Everyday City

3 people use this app

ABOUT THIS APP

Who can see posts this app makes for you on your Facebook timeline: [?]

Everyone

THIS APP WILL RECEIVE:

- Your basic info [?]
- Your email address (aaron@parecki.com)
- Your location

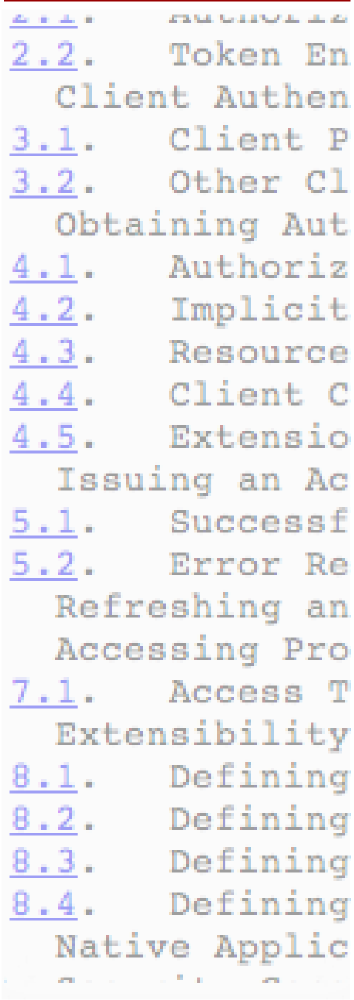
By proceeding, you will be taken to everydaycity.com · Report App

On success, user is redirected back to your site with the access token in the fragment

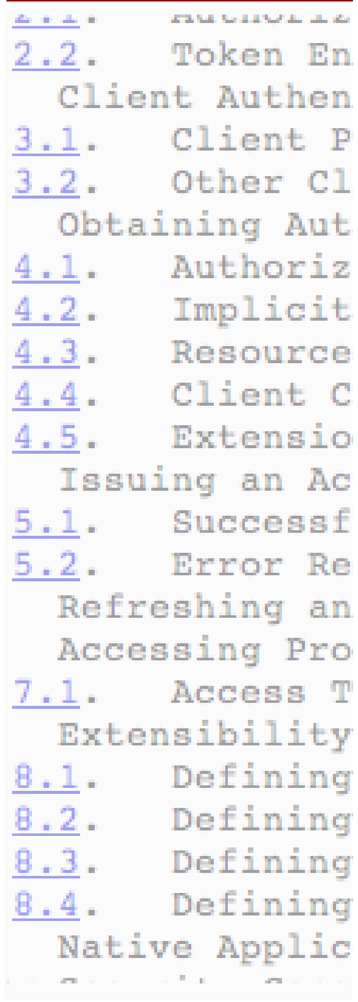
https://example.com/auth**#token=ACCESS_TOKEN**

On error, user is redirected back to your site with error code

https://example.com/auth#error=access_denied



2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

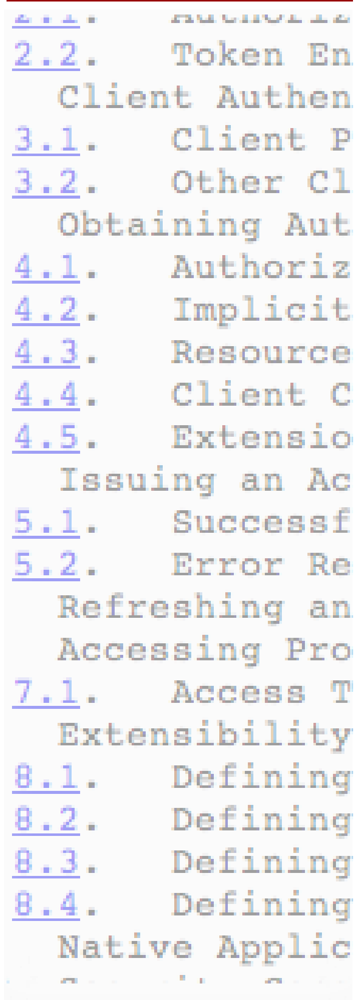

2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

Browser-Based Apps

- Use the “Implicit” grant type
- No server-side code needed
- Client secret not used
- Browser makes API requests directly

Username/Password

Password Grant


2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

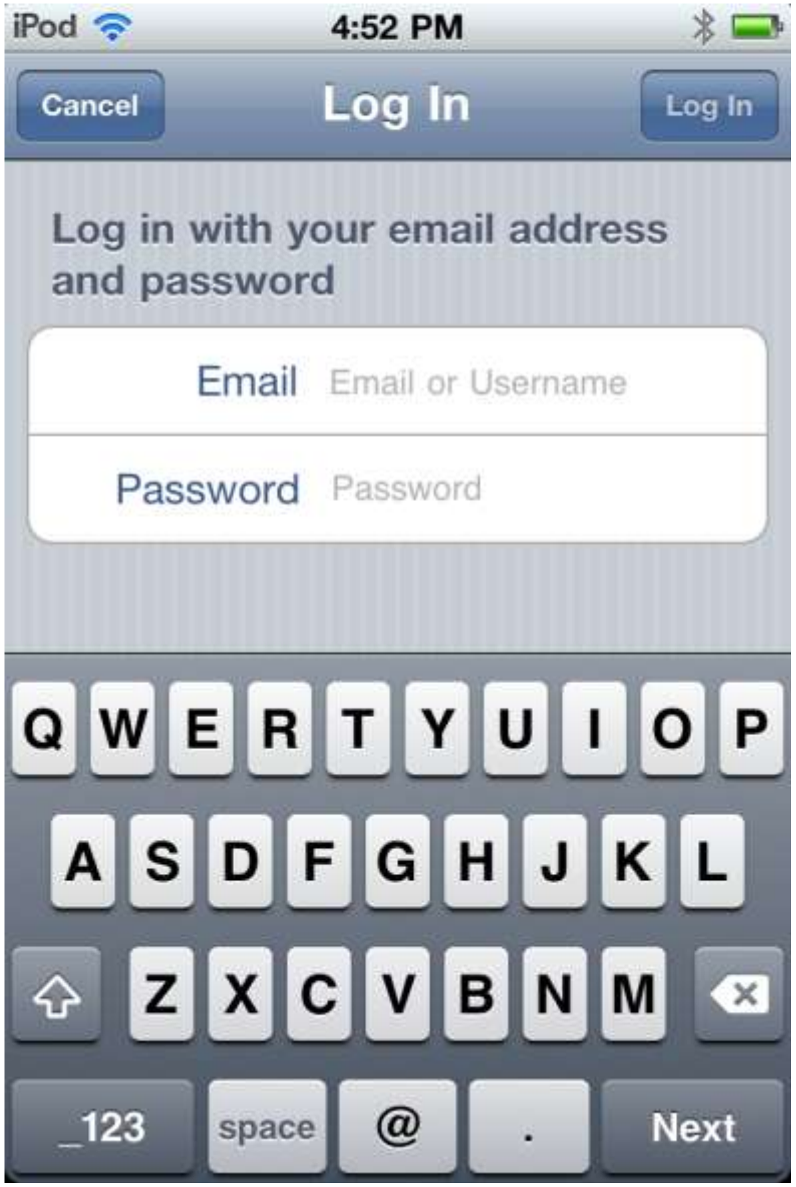
Password Grant

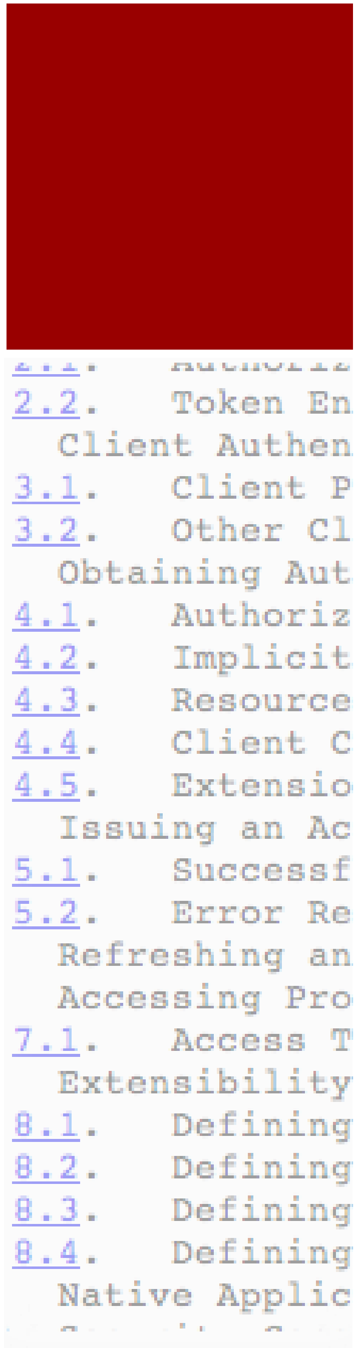
Password grant is only appropriate for trusted clients, most likely first-party apps only.

If you build your own website as a client of your API, then this is a great way to handle logging in.

Password Grant Type

Only appropriate for your service's website or your service's mobile apps.





Password Grant

```
POST https://api.example.com/oauth/token
```

```
Post Body:
```
`grant_type=password`
`&username=USERNAME`
`&password=PASSWORD`
```
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

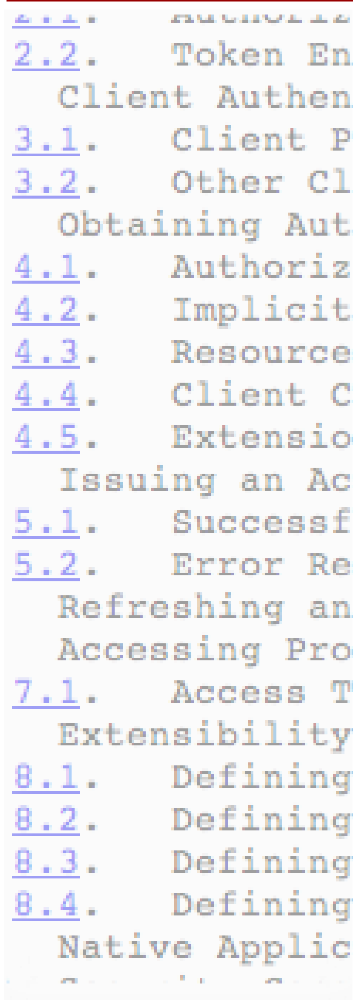
Response:

```
{
  "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"e1qoXg7Ik2RRua48lXIV"
}
```

Application Access

Client Credentials Grant

Client Credentials Grant



```
2.2.    Token En
  Client Authen
3.1.    Client P
3.2.    Other Cl
  Obtaining Aut
4.1.    Authoriz
4.2.    Implicit
4.3.    Resource
4.4.    Client C
4.5.    Extensio
  Issuing an Ac
5.1.    Successf
5.2.    Error Re
  Refreshing an
  Accessing Pro
7.1.    Access T
  Extensibility
8.1.    Defining
8.2.    Defining
8.3.    Defining
8.4.    Defining
  Native Applic
```

```
POST https://api.example.com/1/oauth/token
```

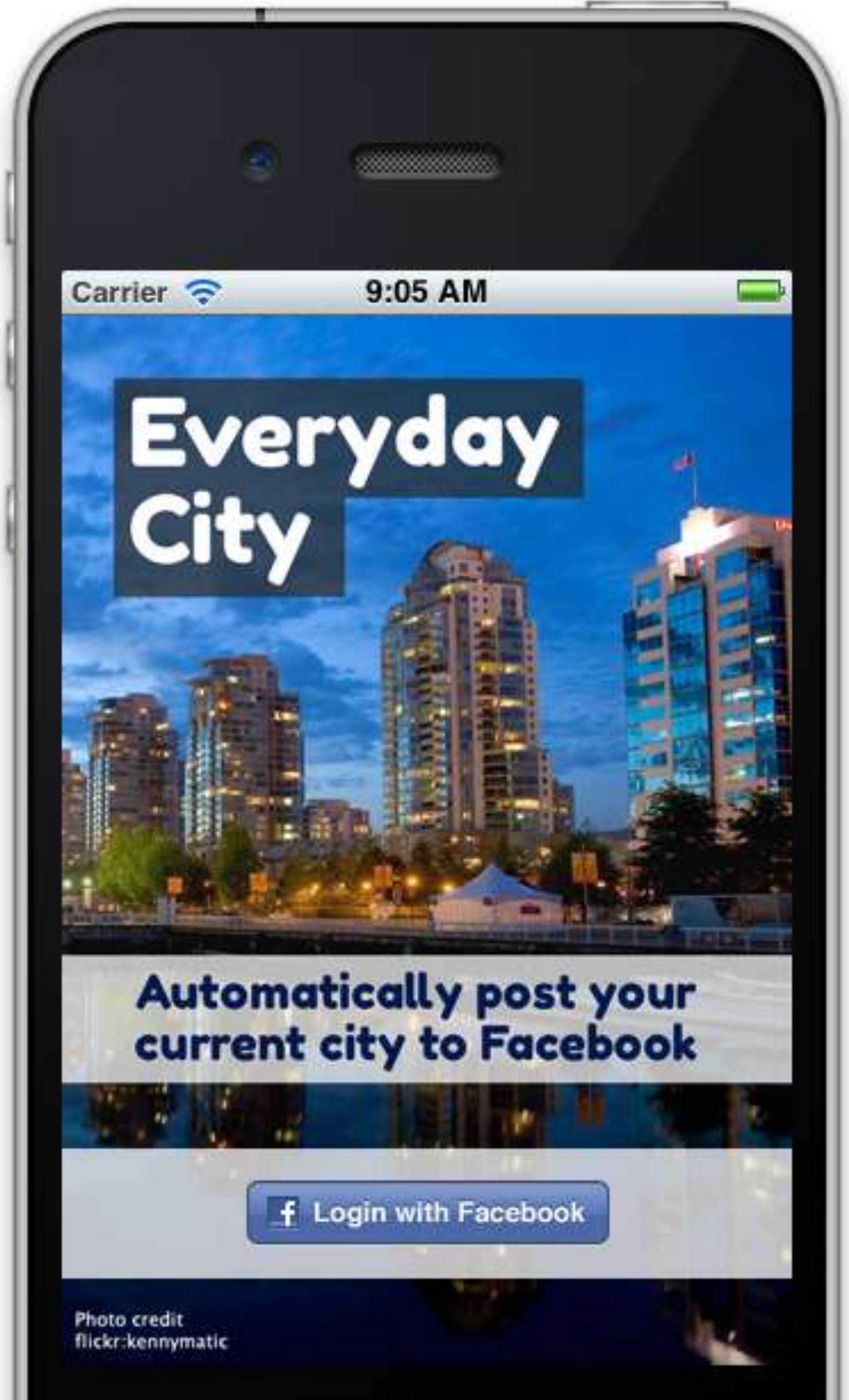
```
Post Body:
```
`grant_type=client_credentials`
```
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

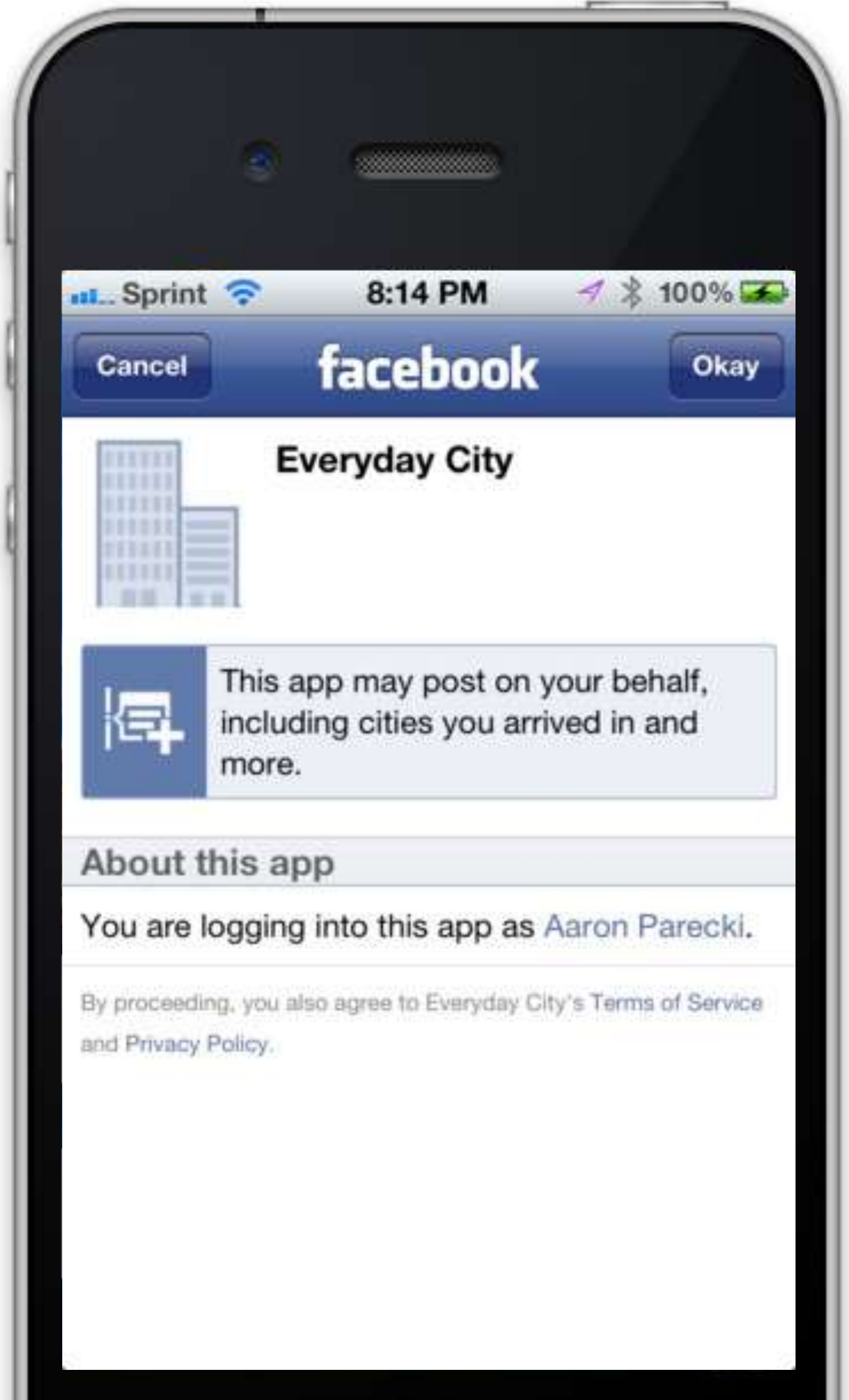
Response:

```
{
  "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"e1qoXg7Ik2RRua48lXIV"
}
```

Mobile Apps

Implicit Grant


Carrier
9:05 AM
Everyday City
Automatically post your current city to Facebook
Login with Facebook
Photo credit
flickr:kennymatic


Sprint 8:14 PM 100%

Cancel facebook Okay

Everyday City

This app may post on your behalf, including cities you arrived in and more.

About this app

You are logging into this app as Aaron Parecki.

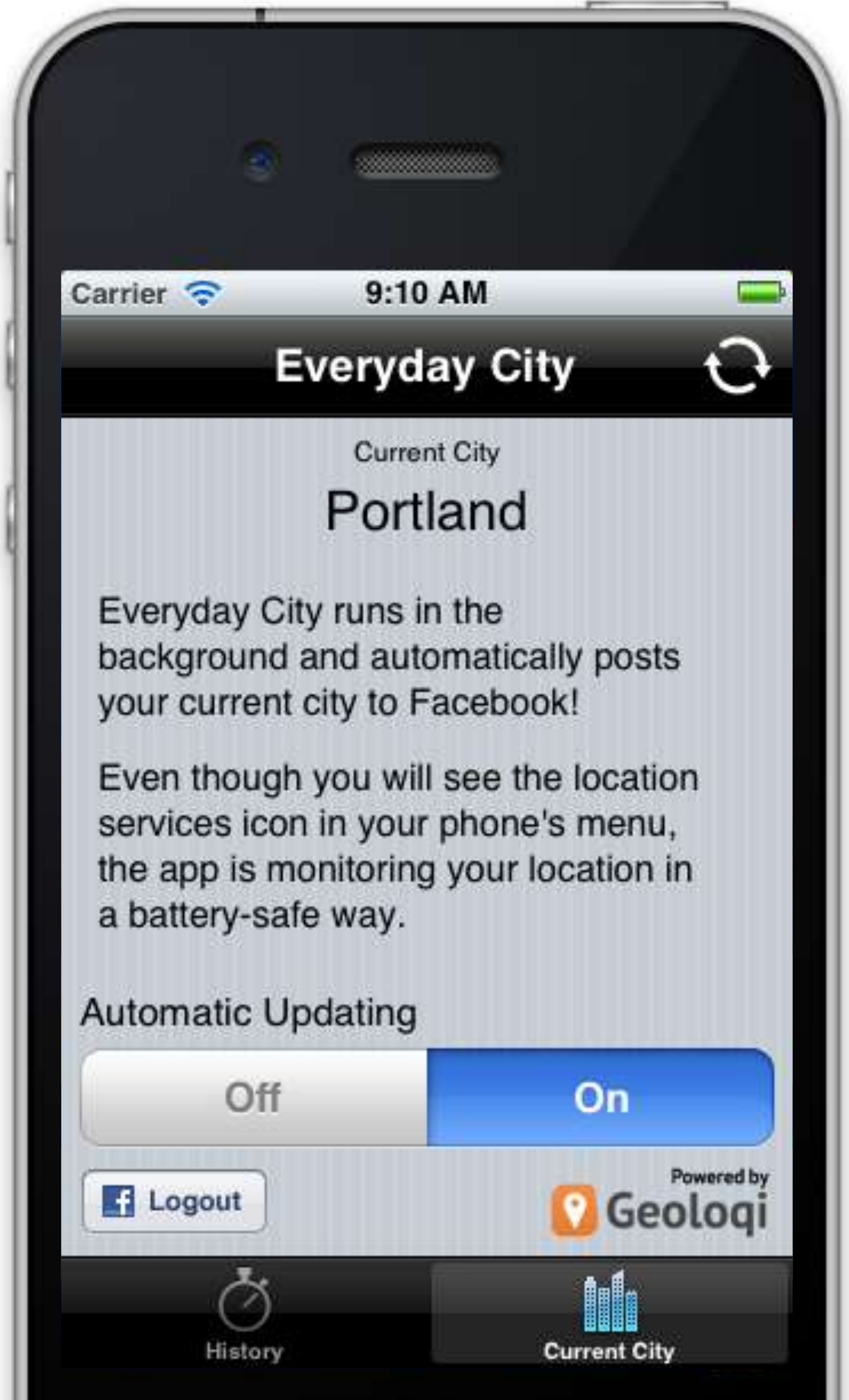
By proceeding, you also agree to Everyday City's Terms of Service and Privacy Policy.

Redirect back to your app

Facebook app redirects back to your app using a custom URI scheme.

Access token is included in the redirect, just like browser-based apps.

```
fb2865://authorize/#access_token=BAAEEmo2nocQBAFFOeRTd
```



Carrier 9:10 AM

Everyday City

Current City

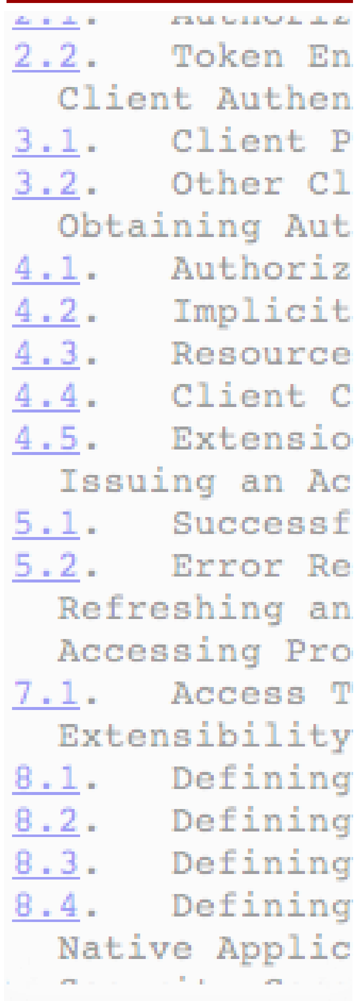
Portland

Everyday City runs in the background and automatically posts your current city to Facebook!

Even though you will see the location services icon in your phone's menu, the app is monitoring your location in a battery-safe way.

Automatic Updating

Off [✓ On]



2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

Mobile Apps

- Use the “Implicit” grant type
- No server-side code needed
- Client secret not used
- Mobile app makes API requests directly

Grant Type Summary

- `authorization_code:`
  Web-server apps
- `implicit:`
  Mobile and browser-based apps
- `password:`
  Username/password access
- `client_credentials:`
  Application access

Grant Types & Response Types

- authorization_code:

  `response_type=code`

- implicit:

  `response_type=token`

Grant Type Review

Authorization Code

- User visits auth page

  `response_type=code`

- User is redirected to your site with auth code

  `http://example.com/?`**`code=xxxxxxx`**

- Your server exchanges auth code for access token

  ```
  POST /token
  code=xxxxxxx&grant_type=authorization_code
  ```

Implicit

- User visits auth page
  `response_type=token`
- User is redirected to your site with access token
  `http://example.com/`**`#token=xxxxxxx`**
- Token is only available to the browser since it’s in the fragment

Password

- Your server exchanges username/password for access token

```
POST /token
username=xxxxxxx&password=yyyyyyy&
grant_type=password
```

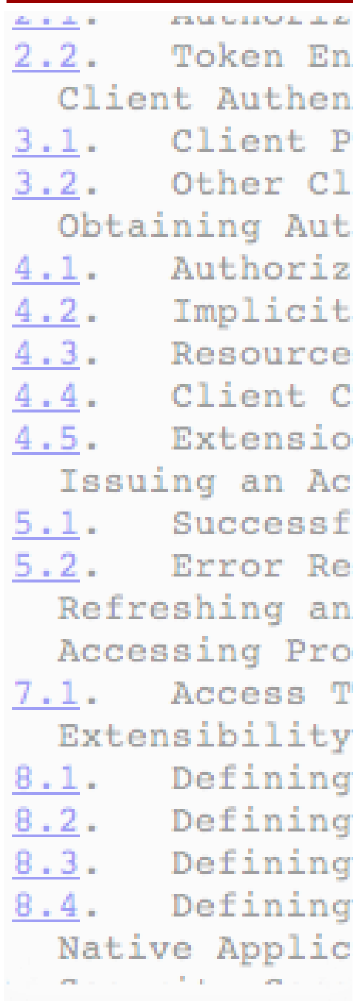
Client Credentials

- Your server exchanges client ID/secret for access token

```
POST /token
client_id=xxxxxxx&client_secret=yyyyyyy&
grant_type=client_credentials
```

Accessing Resources

So you have an access token.
Now what?


2.2. Token En
Client Authen
3.1. Client P
3.2. Other Cl
Obtaining Aut
4.1. Authoriz
4.2. Implicit
4.3. Resource
4.4. Client C
4.5. Extensio
Issuing an Ac
5.1. Successf
5.2. Error Re
Refreshing an
Accessing Pro
7.1. Access T
Extensibility
8.1. Defining
8.2. Defining
8.3. Defining
8.4. Defining
Native Applic

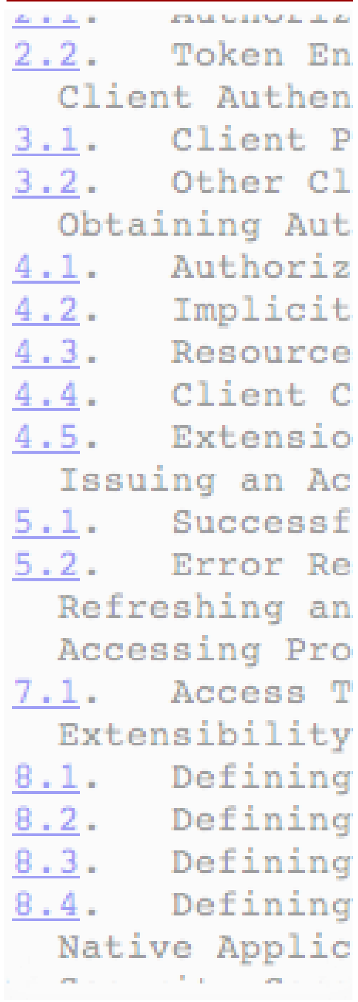
Use the access token to make requests

Now you can make requests using the access token.

```
GET https://api.example.com/me
Authorization: Bearer RsT5OjbzRn430zqMLgV3Ia
```

Access token can be in an HTTP header or a query string parameter

```
https://api.example.com/me?access_token=RsT5Ojb
zRn430zqMLgV3Ia
```



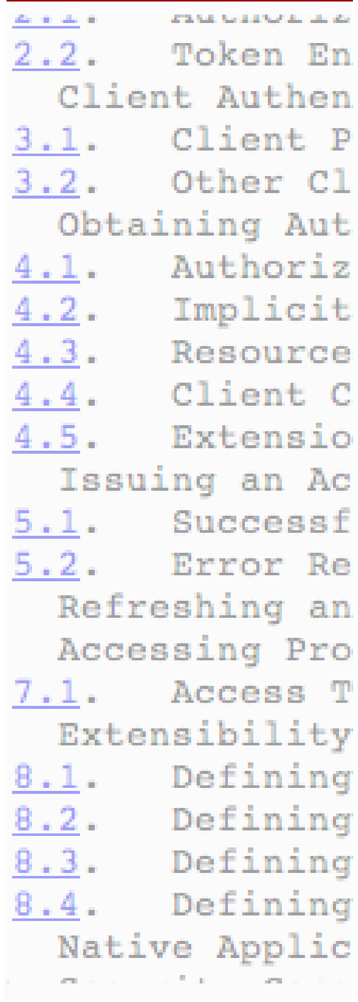
Eventually the access token will expire

When you make a request with an expired token, you will get this response

```
{
  "error":"expired_token"
}
```

Now you need to get a new access token!

Get a new access token using a refresh token



Your server makes the following request

```
POST https://api.example.com/oauth/token

grant_type=refresh_token
&reresh_token=e1qoXg7Ik2RRua48lXIV
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

Your server gets a similar response as the original call to oauth/token with new tokens.

```
{
  "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "expires_in":3600,
  "refresh_token":"e1qoXg7Ik2RRua48lXIV"
}
```

Moving access into separate specs

Bearer tokens vs MAC authentication

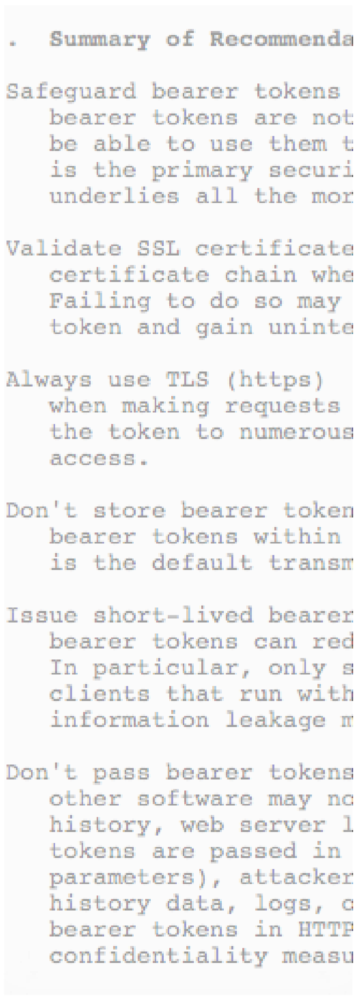
Bearer Tokens

```
GET /1/profile HTTP/1.1
Host: api.example.com
Authorization: Bearer B2mpLsHWhuVFw3YeLFW3f2
```

Bearer tokens are a cryptography-free way to access protected resources.

Relies on the security present in the HTTPS connection, since the request itself is not signed.

Security Recommendations for Clients Using Bearer Tokens



```
.   Summary of Recommenda

Safeguard bearer tokens
    bearer tokens are not
    be able to use them t
    is the primary securi
    underlies all the mor

Validate SSL certificate
    certificate chain whe
    Failing to do so may
    token and gain uninte

Always use TLS (https)
    when making requests
    the token to numerous
    access.

Don't store bearer token
    bearer tokens within
    is the default transm

Issue short-lived bearer
    bearer tokens can red
    In particular, only s
    clients that run with
    information leakage m

Don't pass bearer tokens
    other software may no
    history, web server l
    tokens are passed in
    parameters), attacker
    history data, logs, o
    bearer tokens in HTTP
    confidentiality measu
```

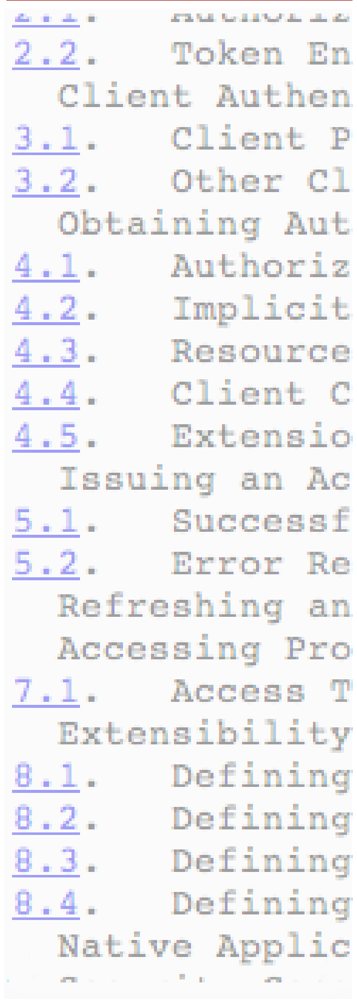
- Safeguard bearer tokens
- Validate SSL certificates
- Always use https
- Don’t store bearer tokens in plaintext cookies
- Issue short-lived bearer tokens
- Don’t pass bearer tokens in page URLs

MAC Tokens

```
GET /1/profile HTTP/1.1
Host: api.example.com
Authorization: MAC id="jd93dh9dh39D",
                   nonce="273156:di3hvdf8",
                   bodyhash="k9kbtCIyI3/FEfpS/oIDjk6k=",
                   mac="W7bdMZbv9UWOTadASIQHagZyirA="
```

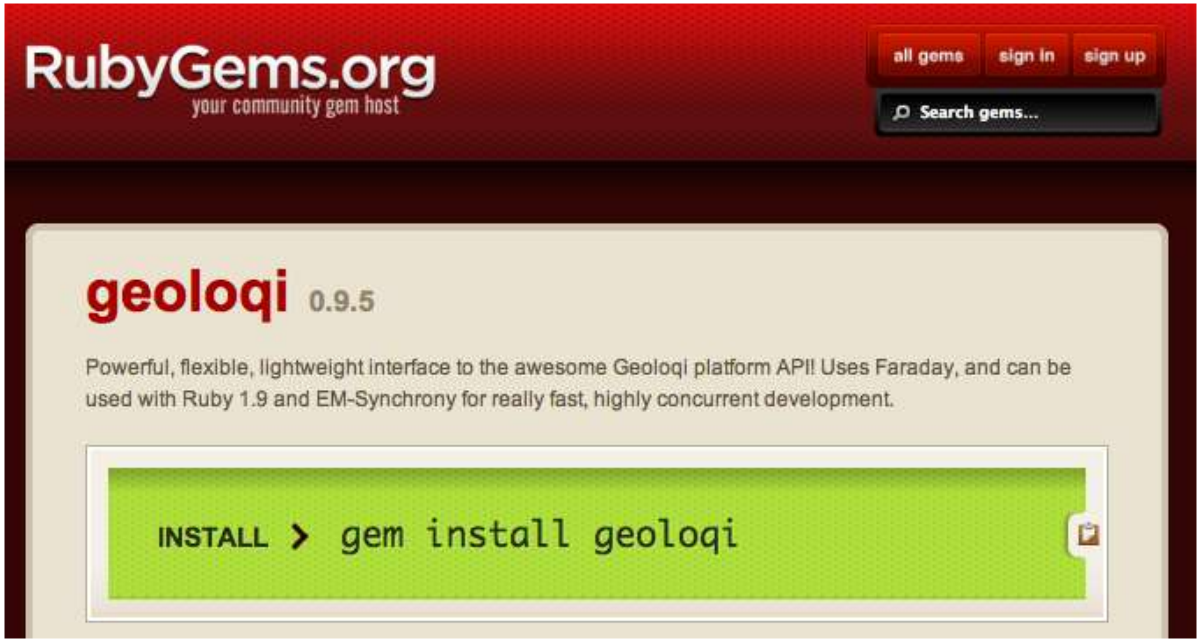
MAC tokens provide a way to make authenticated requests with cryptographic verification of the request.

Similar to the original OAuth 1.0 method of using signatures.



OAuth 2 Clients

Client libraries should handle refreshing the token automatically behind the scenes.



geoloqi 0.9.5

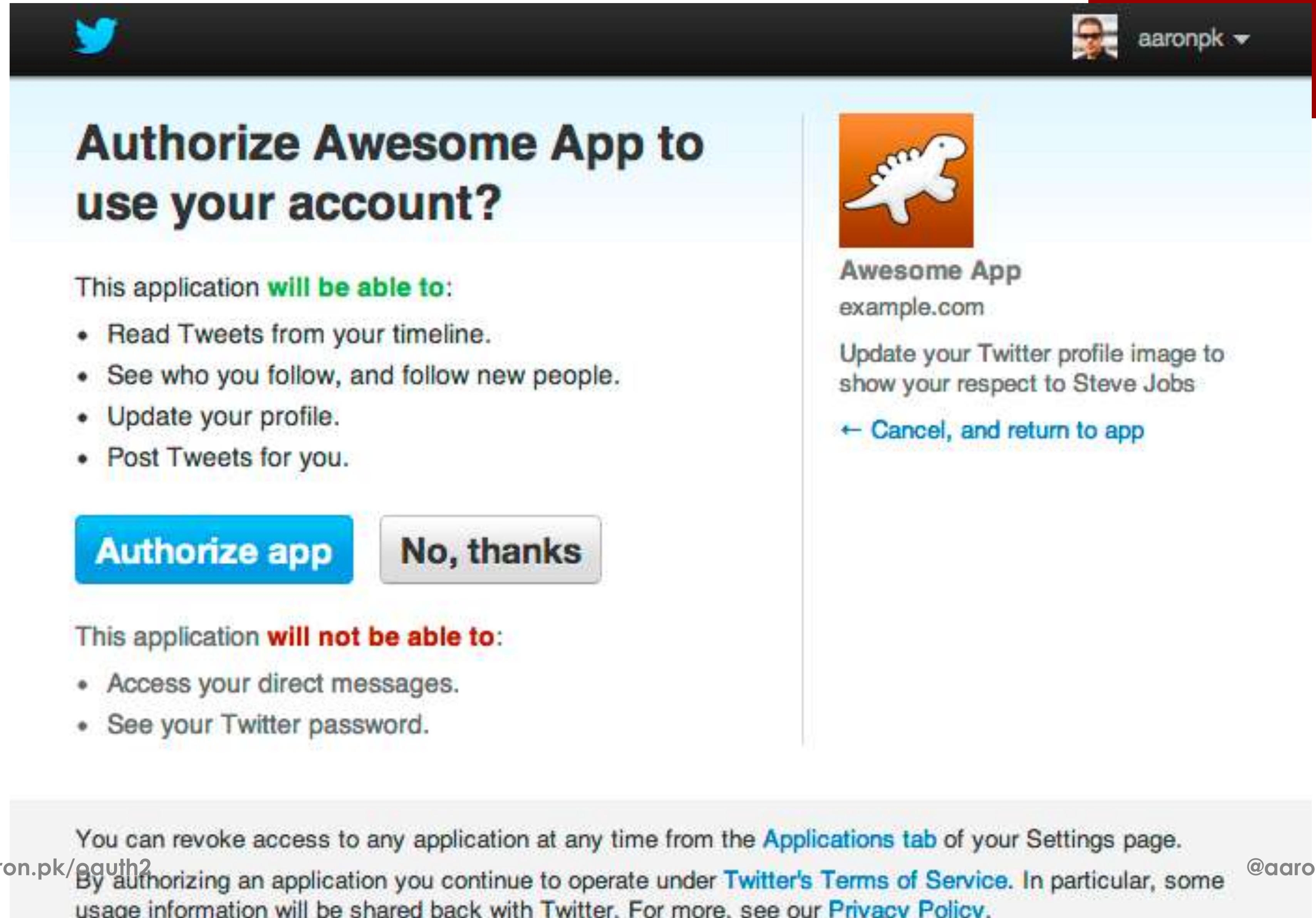
Powerful, flexible, lightweight interface to the awesome Geoloqi platform API! Uses Faraday, and can be used with Ruby 1.9 and EM-Synchrony for really fast, highly concurrent development.

```
INSTALL > gem install geoloqi
```

Scope

Limiting access to resouces

Limiting Access to Third Parties



aaronpk

Authorize Awesome App to use your account?

This application **will be able to**:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

This application **will not be able to**:

- Access your direct messages.
- See your Twitter password.

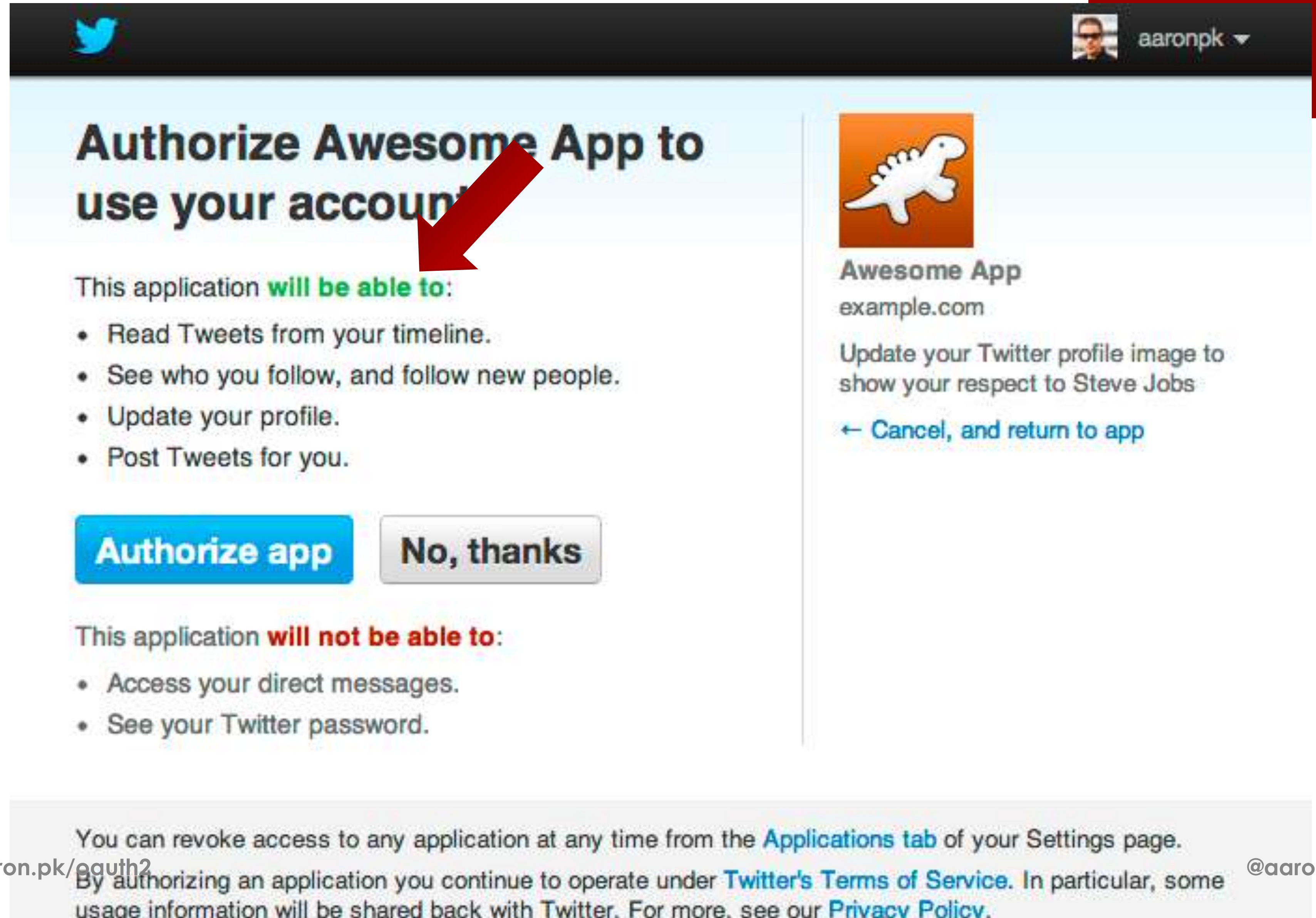
Awesome App
example.com
Update your Twitter profile image to show your respect to Steve Jobs

← Cancel, and return to app

You can revoke access to any application at any time from the Applications tab of your Settings page.

By authorizing an application you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Limiting Access to Third Parties



aaronpk

Authorize Awesome App to use your account?

This application **will be able to**:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

This application **will not be able to**:

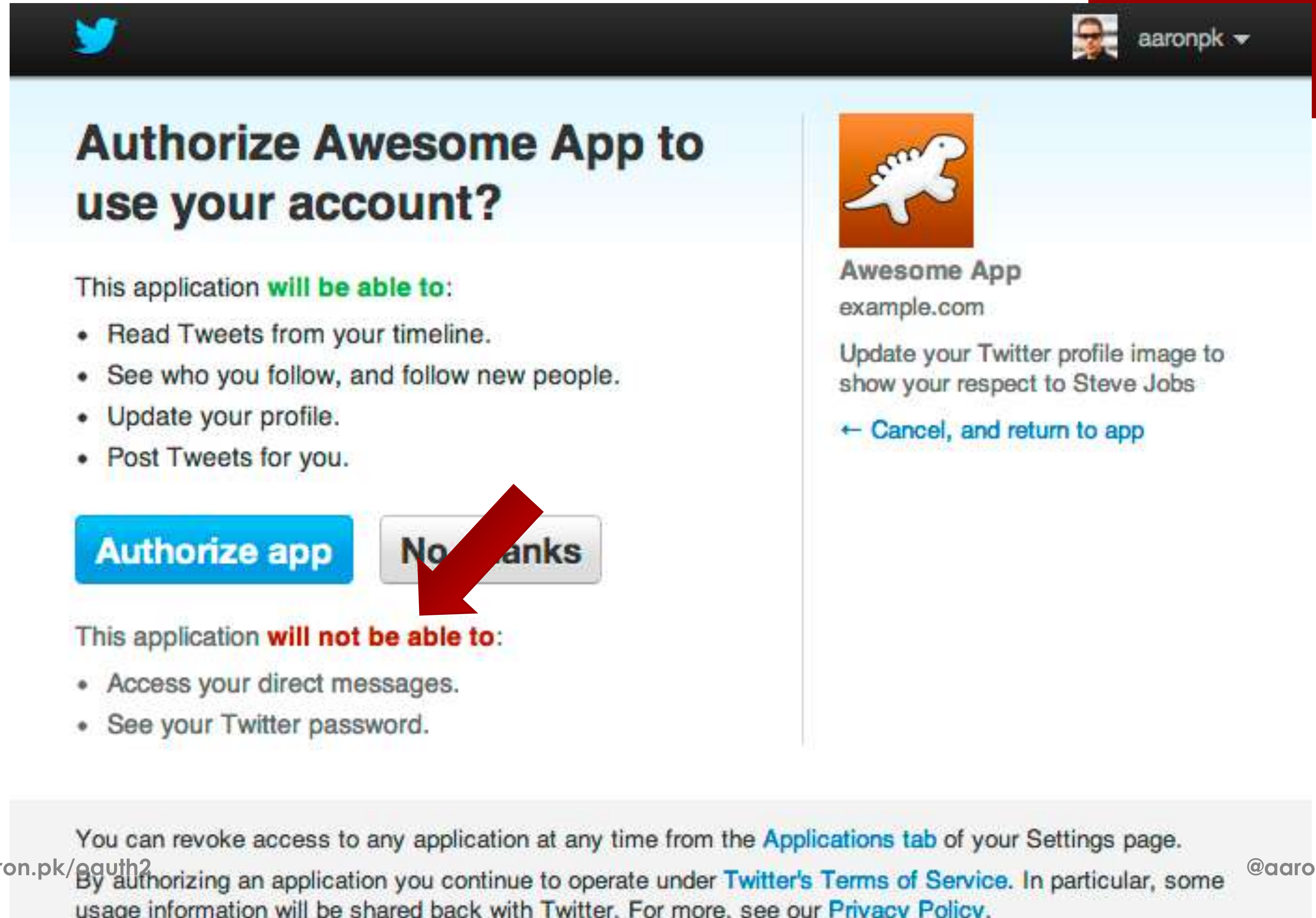
- Access your direct messages.
- See your Twitter password.

Awesome App
example.com
Update your Twitter profile image to show your respect to Steve Jobs

← Cancel, and return to app

You can revoke access to any application at any time from the Applications tab of your Settings page.
By authorizing an application you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Limiting Access to Third Parties



aaronpk

Authorize Awesome App to use your account?

This application **will be able to**:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

This application **will not be able to**:

- Access your direct messages.
- See your Twitter password.

Awesome App
example.com

Update your Twitter profile image to show your respect to Steve Jobs

← Cancel, and return to app

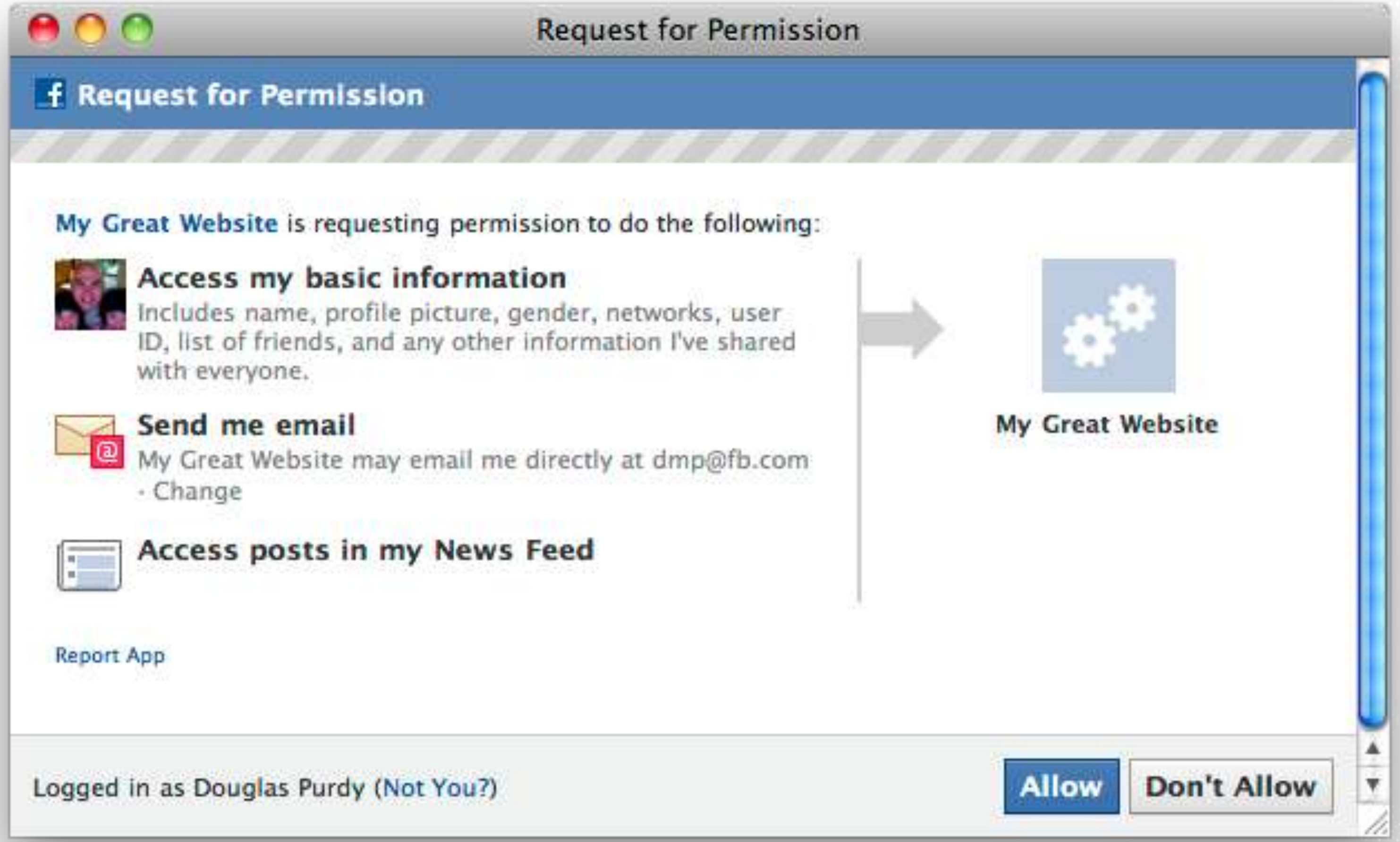
You can revoke access to any application at any time from the Applications tab of your Settings page.
By authorizing an application you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

OAuth 2 `scope`

- Created to limit access to the third party.
- The scope of the access request expressed as a list of space-delimited strings.
  - In practice, many people use comma-separators instead.
- The spec does not define any values, it's left up to the implementor.
- If the value contains multiple strings, their order does not matter, and each string adds an additional access range to the requested scope.

OAuth 2 **scope** on Facebook

```
https://www.facebook.com/dialog/oauth?
client_id=YOUR_APP_ID&redirect_uri=YOUR_URL
&scope=email,read_stream
```



Request for Permission

Request for Permission

My Great Website is requesting permission to do the following:

Access my basic information
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

Send me email
My Great Website may email me directly at dmp@fb.com · Change

Access posts in my News Feed

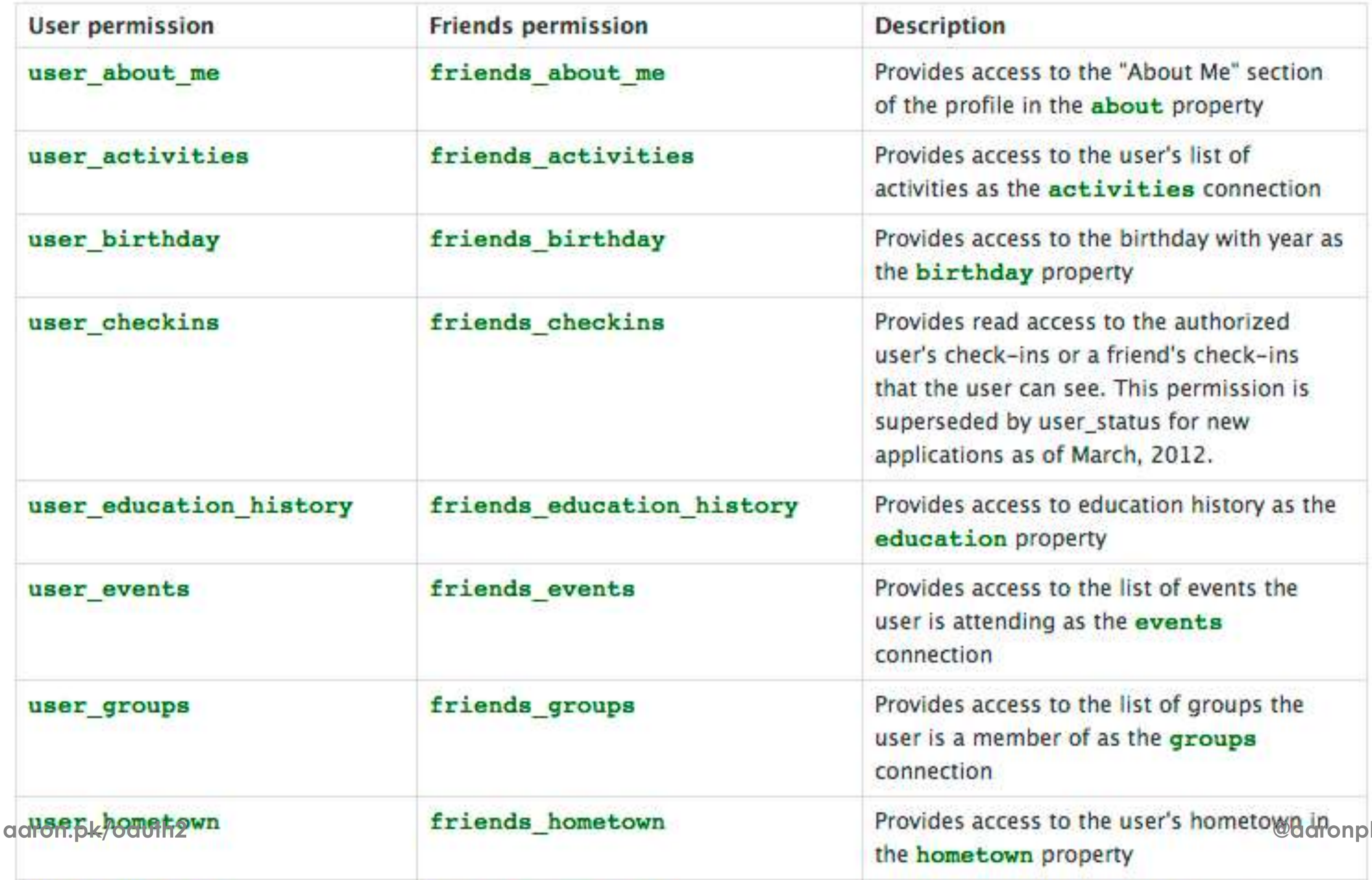
My Great Website

Report App

Logged in as Douglas Purdy (Not You?)

Allow | Don't Allow

OAuth 2 **scope** on Facebook



User permission	Friends permission	Description
user_about_me	friends_about_me	Provides access to the "About Me" section of the profile in the **about** property
user_activities	friends_activities	Provides access to the user's list of activities as the **activities** connection
user_birthday	friends_birthday	Provides access to the birthday with year as the **birthday** property
user_checkins	friends_checkins	Provides read access to the authorized user's check-ins or a friend's check-ins that the user can see. This permission is superseded by user_status for new applications as of March, 2012.
user_education_history	friends_education_history	Provides access to education history as the **education** property
user_events	friends_events	Provides access to the list of events the user is attending as the **events** connection
user_groups	friends_groups	Provides access to the list of groups the user is a member of as the **groups** connection
user_hometown	friends_hometown	Provides access to the user's hometown in the **hometown** property

OAuth 2 **`scope`** on Github

```
https://github.com/login/oauth/authorize?
  client_id=...&scope=user,public_repo
```

`user`

- Read/write access to profile info only.

`public_repo`

- Read/write access to public repos and organizations.

`repo`

- Read/write access to public and private repos and organizations.

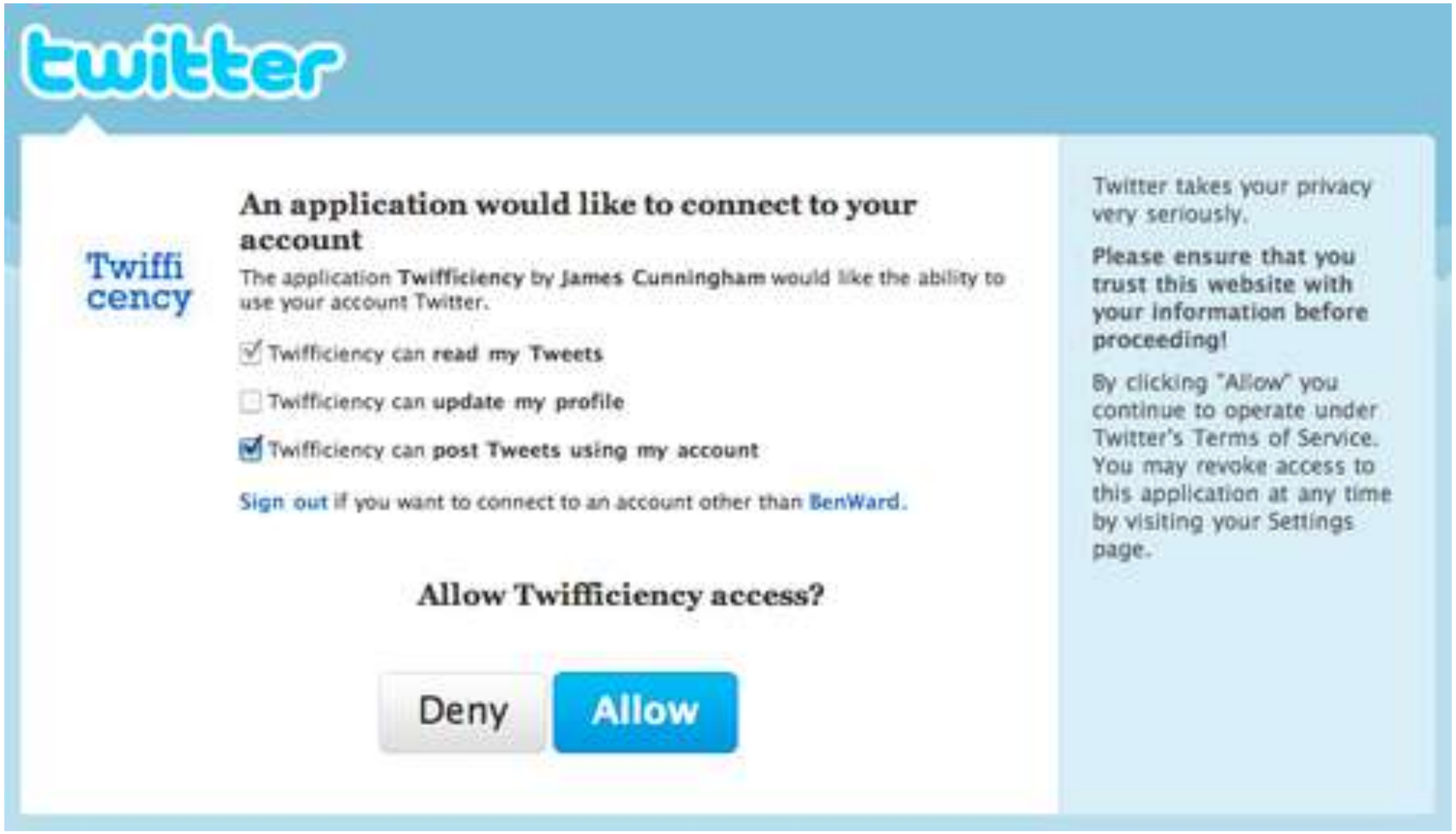
`delete_repo`

- Delete access to adminable repositories.

`gist`

- write access to gists.

Proposed New UI for Twitter
by Ben Ward



An application would like to connect to your account

The application **Twifficiency** by **James Cunningham** would like the ability to use your account Twitter.

- [✓] Twifficiency can **read my Tweets**
- [ ] Twifficiency can **update my profile**
- [✓] Twifficiency can **post Tweets using my account**

Sign out if you want to connect to an account other than BenWard.

Allow Twifficiency access?

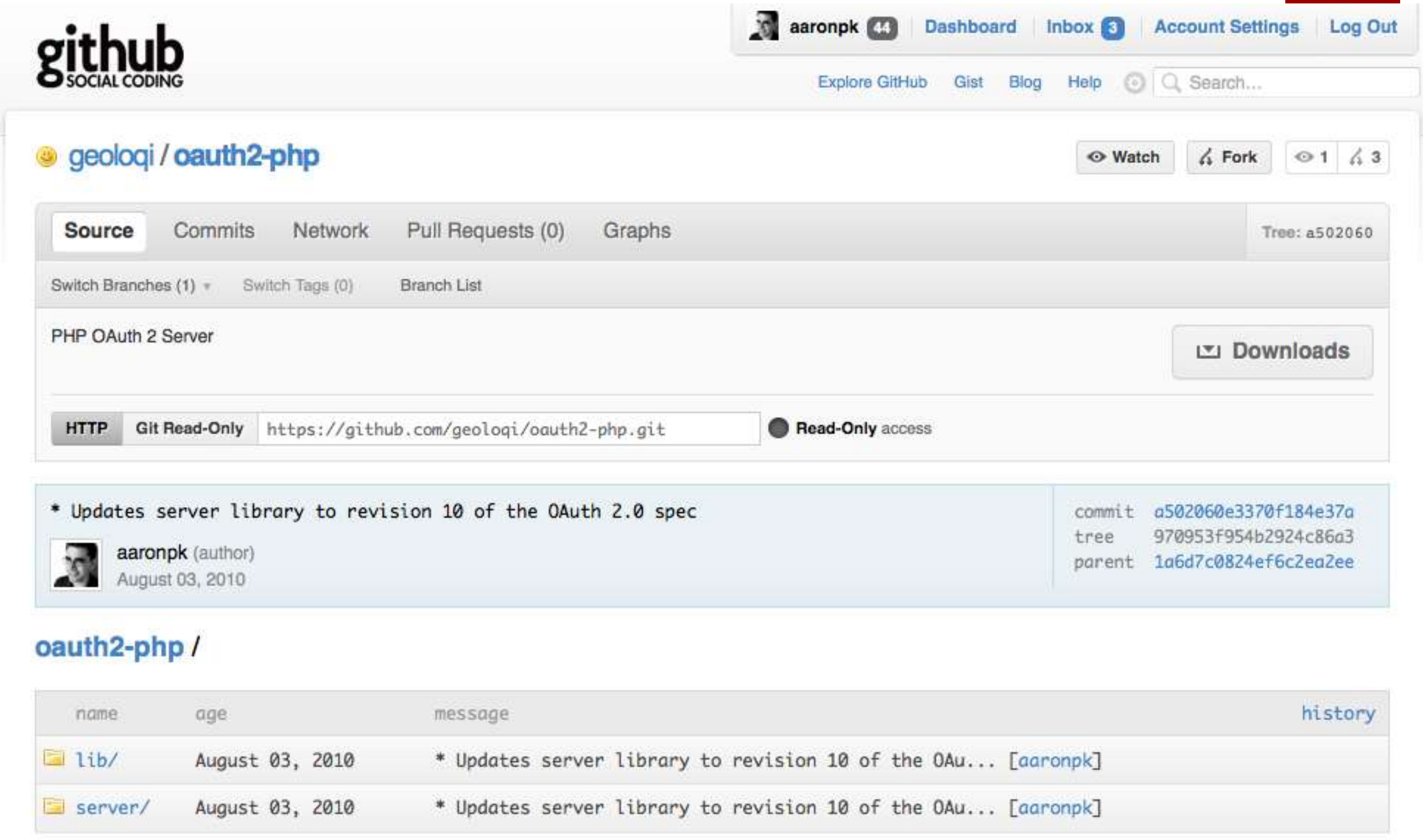
Twitter takes your privacy very seriously.

Please ensure that you trust this website with your information before proceeding!

By clicking "Allow" you continue to operate under Twitter's Terms of Service. You may revoke access to this application at any time by visiting your Settings page.

http://blog.benward.me/post/968515729

Implementing an OAuth Server



geoloqi / oauth2-php

Tree: a502060

PHP OAuth 2 Server

https://github.com/geoloqi/oauth2-php.git

Read-Only access

```
* Updates server library to revision 10 of the OAuth 2.0 spec
```

aaronpk (author)
August 03, 2010

```
commit  a502060e3370f184e37a
tree    970953f954b2924c86a3
parent  1a6d7c0824ef6c2ea2ee
```

oauth2-php /

```
name      age              message                                                    history
lib/      August 03, 2010  * Updates server library to revision 10 of the OAu... [aaronpk]
server/   August 03, 2010  * Updates server library to revision 10 of the OAu... [aaronpk]
```

Implementing an OAuth Server

- Find a server library already written:
  - A short list available here: http://oauth.net/2/
- Read the spec of your chosen draft, *in its entirety*.
  - These people didn't write the spec for you to ignore it.
  - Each word is chosen carefully.
- Ultimately, each implementation is somewhat different, since in many cases the spec says SHOULD and leaves the choice up to the implementer.
- Understand the security implications of the implementation choices you make.

Implementing an OAuth Server

- Choose which grant types you want to support
  - Authorization Code – for traditional web apps
  - Implicit – for browser-based apps and mobile apps
  - Password – for your own website or mobile apps
  - Client Credentials – if applications can access resources on their own
- Choose whether to support Bearer tokens, MAC or both
- Define appropriate scopes for your service

OAuth 2 **`scope`** on your service

- Think about what scopes you might offer
- Don't over-complicate it for your users
- Read vs write is a good start

Mobile Applications

- External user agents are best
  - Use the service's primary app for authentication, like Facebook
  - Or open native Safari on iPhone rather than use an embedded browser
- Auth code or implicit grant type
  - In both cases, the client secret should never be used, since it is possible to decompile the app which would reveal the secret

Staying Involved

Join the Mailing List!

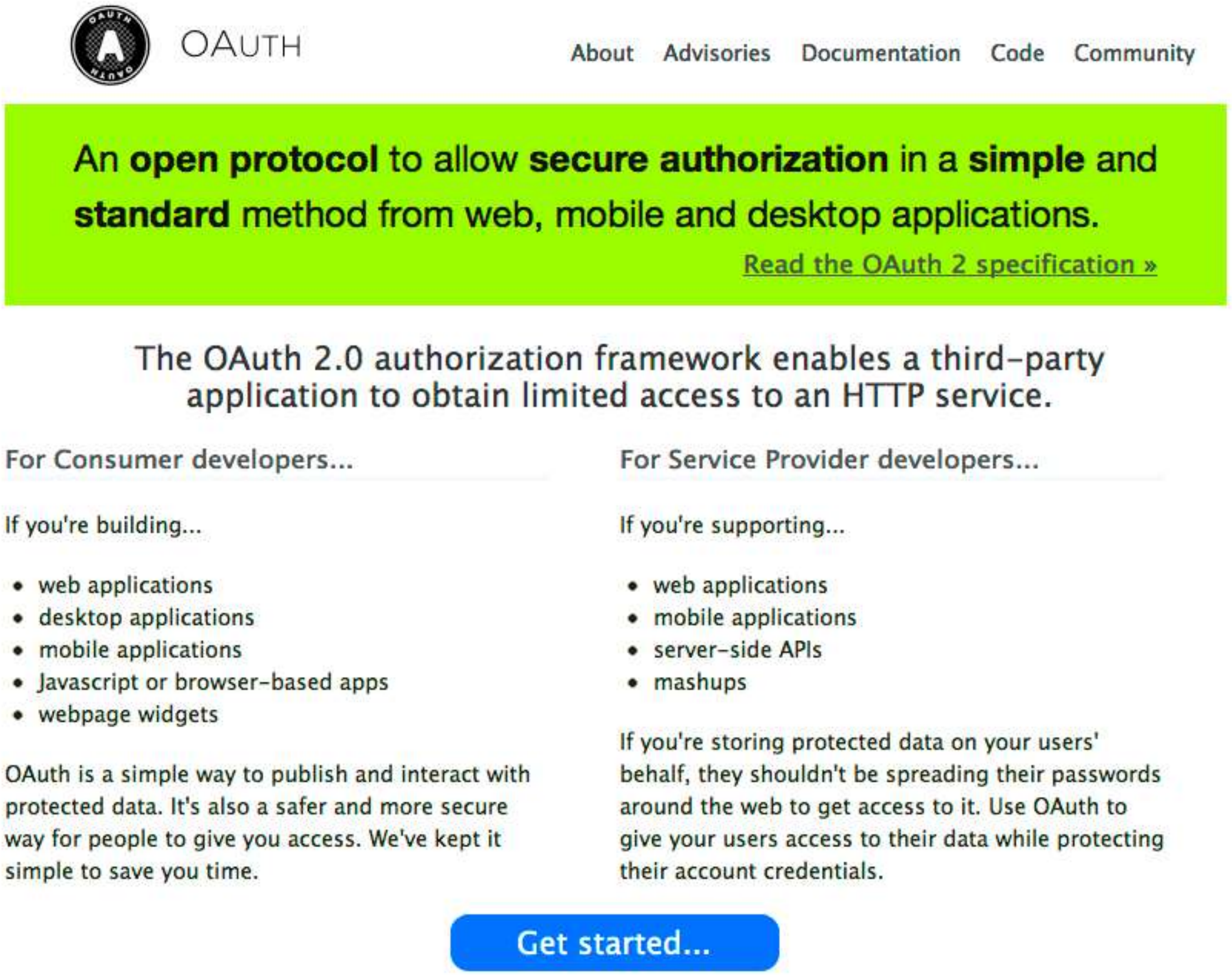
- https://www.ietf.org/mailman/listinfo/oauth
- People talk about OAuth
- Keep up to date on changes
- People argue about OAuth
- It's fun!

oauth Discussion Archive - D

[Prev Page] [Next Page] [Thread Index] [IETF Mailing List D

- Feb 03 2011
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Hum about 'Removal: OAuth
  - Re: [OAUTH-WG] Hum about 'Removal: HTTP
  - Re: [OAUTH-WG] Bearer token type and scheme
  - Re: [OAUTH-WG] Hum about 'Removal: OAuth
  - Re: [OAUTH-WG] Bearer token type and scheme
  - [OAUTH-WG] New Working Group Items?, *Han*
  - Re: [OAUTH-WG] Hum about 'Removal: OAuth
  - Re: [OAUTH-WG] Hum about 'Removal: HTTP
  - Re: [OAUTH-WG] Hum about 'Removal: OAuth
  - Re: [OAUTH-WG] Hum about 'Removal: OAuth
  - Re: [OAUTH-WG] Hum about 'Removal: HTTP
  - Re: [OAUTH-WG] Hum about 'Removal: HTTP
  - [OAUTH-WG] JSON Cryptographic Syntax and
  - Re: [OAUTH-WG] Hum about 'Removal: HTTP
  - [OAUTH-WG] Bearer token type and scheme na
  - [OAUTH-WG] Hum about 'Removal: HTTP Basi
  - [OAUTH-WG] Hum about 'Removal: Client Asse
  - [OAUTH-WG] Hum about 'Removal: OAuth2 HT
  - [OAUTH-WG] OAuth version -12 specification,
  - [OAUTH-WG] Token Revocation (was: Re-Chart
- Feb 02 2011
  - [OAUTH-WG] Fwd: I-D Action:draft-ietf-oauth-
  - [OAUTH-WG] I-D Action:draft-ietf-oauth-saml2
- Jan 31 2011
  - Re: [OAUTH-WG] assertion, client_assertion_ty
- Jan 30 2011
  - Re: [OAUTH-WG] Update required for bearer to
  - Re: [OAUTH-WG] assertion, client_assertion_ty
  - Re: [OAUTH-WG] Update required for bearer to
  - [OAUTH-WG] assertion, client_assertion_type a
  - Re: [OAUTH-WG] OAuth 2.0 Bearer Token Spe
  - Re: [OAUTH-WG] Native Client Extension, *Sky*
- Jan 28 2011
  - Re: [OAUTH-WG] Removal: Client Assertion Cr
- Jan 28 2011
  - Re: [OAUTH-WG] Removal: Client Assertion Cr
  - Re: [OAUTH-WG] OAuth 2.0 Bearer Token Spe
  - Re: [OAUTH-WG] OAuth 2.0 Bearer Token Spe
  - Re: [OAUTH-WG] OAuth 2.0 Bearer Token Spe
  - Re: [OAUTH-WG] OAuth 2.0 Bearer Token Spe

oauth.net



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

Read the OAuth 2 specification »

The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service.

For Consumer developers...

If you're building...

- web applications
- desktop applications
- mobile applications
- 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...

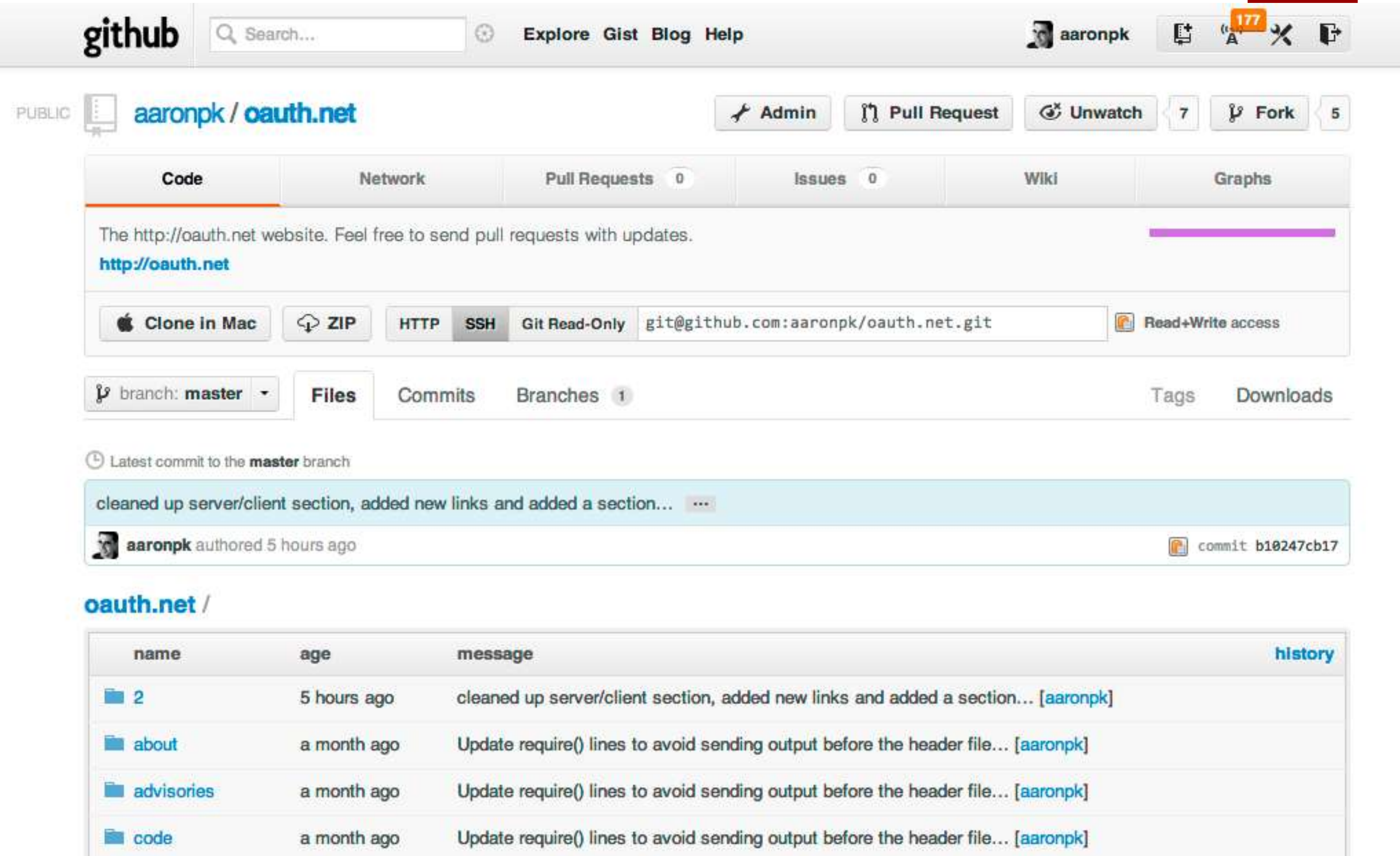
- web applications
- mobile applications
- server-side APIs
- mashups

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.

oauth.net Website

- http://oauth.net
- Source code available on Github
  - github.com/aaronpk/oauth.net
- Please feel free to contribute to the website
- Contribute new lists of libraries, or help update information
- OAuth is community-driven!

github.com/aaronpk/oauth.net



The http://oauth.net website. Feel free to send pull requests with updates.

http://oauth.net

git@github.com:aaronpk/oauth.net.git

Latest commit to the **master** branch

cleaned up server/client section, added new links and added a section... ···

aaronpk authored 5 hours ago   commit b10247cb17

oauth.net /

name	age	message
2	5 hours ago	cleaned up server/client section, added new links and added a section... [aaronpk]
about	a month ago	Update require() lines to avoid sending output before the header file... [aaronpk]
advisories	a month ago	Update require() lines to avoid sending output before the header file... [aaronpk]
code	a month ago	Update require() lines to avoid sending output before the header file... [aaronpk]

More Info, Slides & Code Samples:

aaron.pk/oauth2

Thanks.

Aaron Parecki
@aaronpk
aaronparecki.com
github.com/aaronpk