

STTwitter

Nicolas Seriot
May 22nd, 2014



CocoaHeads Lausanne

AppSec 2012

A man with glasses and a blue long-sleeved shirt is standing on a stage, pointing his right arm towards a screen on the right. He is wearing a red lanyard with a badge. The background is a light-colored wall with a projection screen showing some text and symbols.

Twitter relies on OAuth to control 3rd party clients.

Bad idea! secret tokens are easy to extract.

Leaked consumer secrets can result in API abuses, DoS and session fixation attacks.

HITB 2013 Amsterdam

Send 165+ bytes
in a tweet.

(...)

Crash OS X / iOS
Twitter clients.

@boblord →

demo callback,
39.109513, -76.772979

HITB, April 10th



REUTERS
Reuters Top News ✓
@Reuters
Reuters.com brings you the latest news from around the world, covering breaking news in business, politics, technology, and more.
<http://www.reuters.com/>

84,962 TWEETS 1,056 FOLLOWING 2,733,867 FOLLOWERS

Follow



CNN ✓
@CNN
Bringing you breaking news, the biggest moments from CNN TV, and the stories and videos gaining attention on CNN.com and social media.
<http://www.cnn.com>

31,460 TWEETS 756 FOLLOWING 8,090,009 FOLLOWERS

Follow



AFP ✓
Agence France-Presse ✓
@AFP
News - and the stories behind the news - from AFP's reporters around the world. In French: @afpr. In Spanish: @AFPespanol. For photos: @AFPphoto
France · <http://www.afp.com>

21,711 TWEETS 381 FOLLOWING 63,699 FOLLOWERS

Follow

April 23rd, 2013



AP The Associated Press ✓
@AP

Breaking: Two Explosions in the White House and Barack Obama is injured

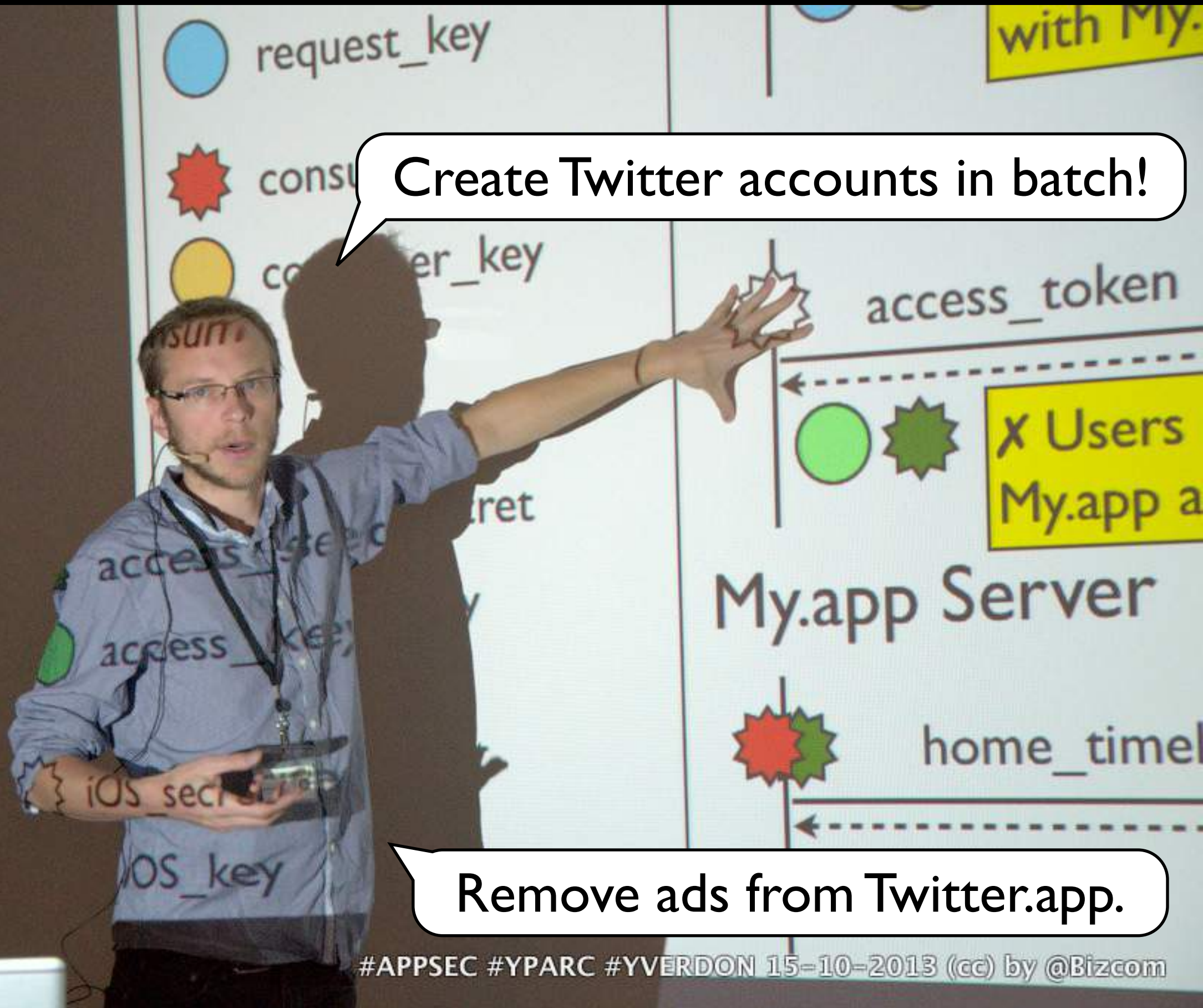
← Reply ↻ Retweet ★ Favorite ⋮ More

2,894 RETWEETS 134 FAVORITES

10:07 AM - 23 Apr 13



AppSec 2013



Abusing Twitter API

Table of Contents

1. [The Old and the New Twitter](#)
2. [Twitter and OAuth](#)
 - 2.1 [PIN-based Authentication](#)
 - 2.2 [xAuth Authentication](#)
 - 2.3 [Application Only Authentication](#)
 - 2.4 [Reverse Authentication](#)
3. [Extracting Consumer Tokens](#)
 - 3.1 [/usr/bin/strings](#)
 - 3.2 [GDB](#)
 - 3.3 [Memory Dump](#)
 - 3.4 [DTrace Probes](#)
 - 3.5 [Code Injection](#)
4. [Finding Undocumented API Endpoints](#)
 - 4.1 [Reverse Engineering Twitter.app](#)
 - 4.2 [Promoted Content](#)
 - 4.3 [Accounts Creation](#)
 - 4.4 [Hardcoded Tokens](#)
5. [Twitter on OS X and iOS](#)
 - 5.1 [Twitter APIs on OS X / iOS Integration](#)
 - 5.2 [STTwitter, an Objective-C Library for Twitter](#)
 - 5.3 [TwitHunter, a Twitter Client over STTwitter](#)
6. [Security Considerations](#)
 - 6.1 [Reaching application limits](#)
 - 6.2 [Bearer token invalidation](#)
 - 6.3 [Keys revocation](#)
 - 6.4 [OAuth Session fixation attack](#)
 - 6.5 [Sending passwords over the network](#)
 - 6.6 [Fighting spam](#)
 - 6.7 [Keys protection](#)
 - 6.8 [Incorrect, downgraded OAuth implementation](#)
7. [Conclusion](#)

http://seriot.ch/abusing_twitter_api.php

This repository Search or type a command Explore Gist Blog Help nst +- ✕

PUBLIC nst / STTwitter Unwatch 36 Unstar 482 Fork 70

A comprehensive Objective-C library for Twitter REST API 1.1 — Edit

588 commits 2 branches 13 releases 8 contributors

branch: master STTwitter / +

removed unnecessary precompiled headers

nst authored an hour ago latest commit 685c0a8458

Art	updated screenshot	7 months ago
STTwitter	updated STHTTPRequest	20 hours ago
Tests	split streaming example into streaming and streaming_with_bot	5 days ago
demo_cli	removed unnecessary precompiled headers	an hour ago
demo_ios	added new command-line interface examples	20 days ago
demo_osx	added accounts_creation into demo_cli	5 days ago
.travis.yml	improved travis build	8 days ago
LICENCE.txt	updated years in license	14 days ago
README.md	added references and description of CLI examples	5 days ago
STTwitter.podspec	preparing tag 0.1.3	20 hours ago

Code

Issues 1

Pull Requests 0

Wiki

Pulse

Graphs

Network

Settings

HTTPS clone URL

https://github.com/nst/STTwitter

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Cool side effect: an Obj-C library

Agenda

1. Access Twitter API

2. Twitter API

3. STTwitter

1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

1.5 Safari

1.6 Reverse Auth

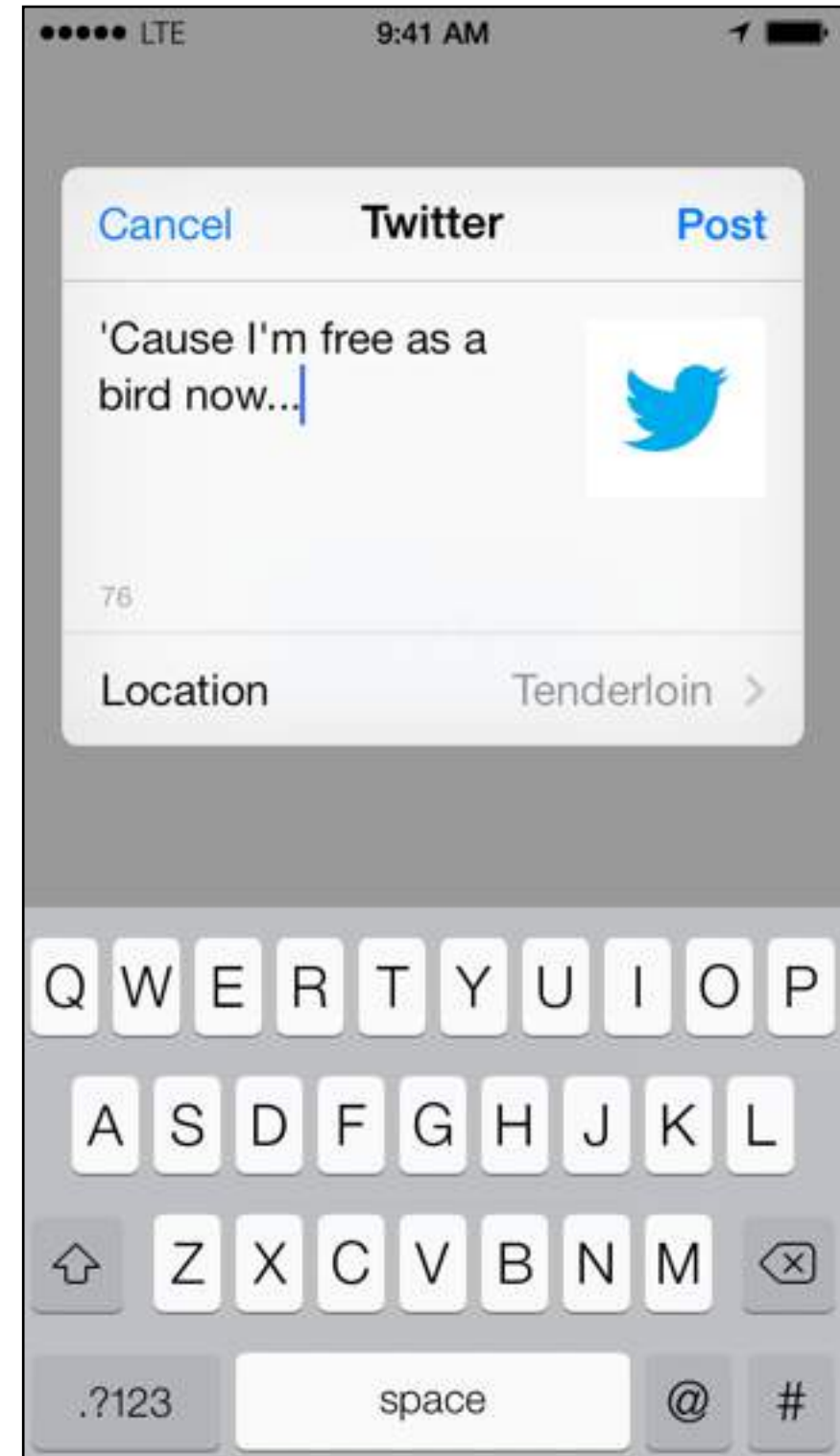
1.7 App Only

2. Twitter API

3. STTwitter

Tweet Sheet

- most simple use case
- when your user want to share contents
- compose a tweet using Setting's account



```
// check Twitter availability
if ([SLComposeViewController isAvailableForServiceType:SLServiceTypeTwitter] == NO)
    return;

// instantiate composer
SLComposeViewController *composeVC =
[SLComposeViewController composeViewControllerForServiceType:SLServiceTypeTwitter];

// add contents
[composeVC setInitialText:@"Test"];
[composeVC addImage:myImage];
[composeVC addURL:myURL];

// set completion handler
[composeVC setCompletionHandler:^(SLComposeViewControllerResult result) {
    if(result == SLComposeViewControllerResultCancelled)
        NSLog(@"post cancelled");
    if(result == SLComposeViewControllerResultDone)
        NSLog(@"post successful");
}];

// present tweet sheet
[self presentViewController:composeVC animated:YES completion:nil];
```


1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

1.5 Safari

1.6 Reverse Auth

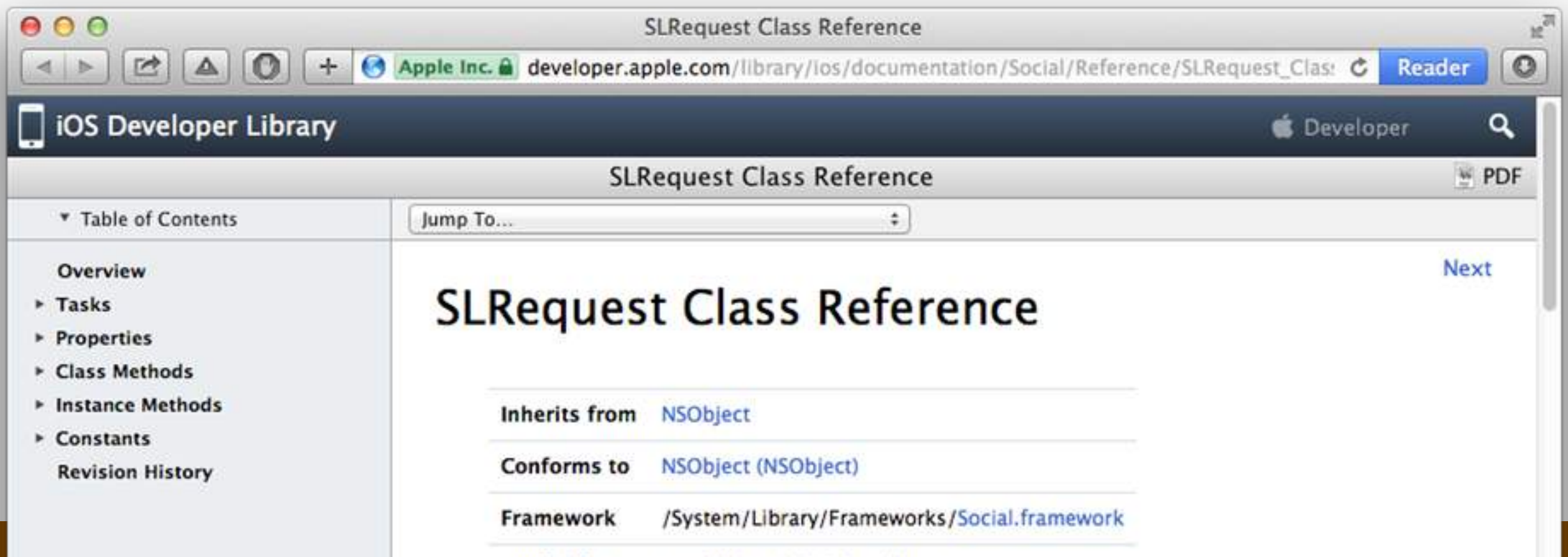
1.7 App Only

2. Twitter API

3. STTwitter

SLRequest

- access the full Twitter API
- create / customise a SLRequest instance
- access the endpoint you need



The screenshot shows a web browser window displaying the Apple Developer Library page for the SLRequest class. The browser's address bar shows the URL: `developer.apple.com/library/ios/documentation/Social/Reference/SLRequest_Clas`. The page title is "SLRequest Class Reference". On the left side, there is a navigation menu with the following items: "Table of Contents", "Overview", "Tasks", "Properties", "Class Methods", "Instance Methods", "Constants", and "Revision History". The main content area displays the class name "SLRequest Class Reference" and its inheritance and framework information:

- Inherits from [NSObject](#)
- Conforms to [NSObject \(NSObject\)](#)
- Framework `/System/Library/Frameworks/Social.framework`

There is a "Jump To..." search box and a "Next" link in the top right corner of the content area.

```

// 1. check that the user has a local Twitter account
if([SLComposeViewController isAvailableForServiceType:SLServiceTypeTwitter] == NO) return;

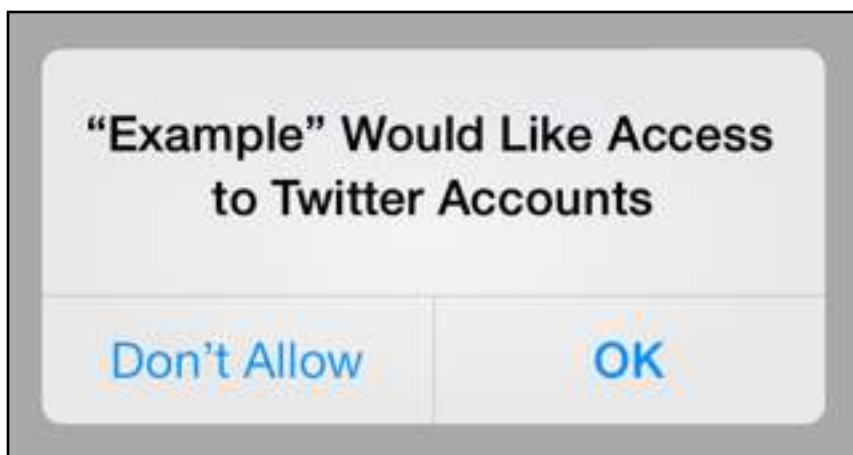
ACAccountStore *accountStore = [[ACAccountStore alloc] init];
ACAccountType *twitterAccountType =
    [accountStore accountTypeWithIdentifier:ACAccountTypeIdentifierTwitter];

// 2. ask the user permission to access his account
[accountStore requestAccessToAccountsWithType:twitterAccountType
    options:nil
    completion:^(BOOL granted, NSError *error) {

    if(granted == NO) {
        NSLog(@"-- error: %@", [error localizedDescription]);
        return;
    };

    // create and execute a request (next slide)
}];

```




```

// 3. create a request
NSURL *url = [NSURL URLWithString:@"https://api.twitter.com/1.1/statuses/user_timeline.json"];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeTwitter
                                requestMethod:SLRequestMethodGET
                                URL:url
                                parameters:@{@"screen_name":@"nst021"}];

// 4. attach an account to the request
NSArray *twitterAccounts = [accountStore accountsWithAccountType:twitterAccountType];
[request setAccount:[twitterAccounts lastObject]];

// 5. execute the request
[request performRequestWithHandler:^(NSData *responseData,
                                   NSHTTPURLResponse *urlResponse,
                                   NSError *error) {

    NSError *jsonError = nil;
    NSDictionary *json = [NSJSONSerialization JSONObjectWithData:responseData
                                                            options:NSJSONReadingAllowFragments
                                                            error:&jsonError];

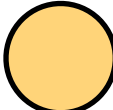
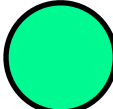

    // caution we're on an arbitrary queue
    [[NSOperationQueue mainQueue] addOperationWithBlock:^(
        NSLog(@"-- json: %@", json);
    )];
}];
}];

```

SLRequest for OAuth

```
GET https://api.twitter.com/1.1/statuses/home_timeline.json
```

```
Authorization = OAuth
```

 **oauth_consumer_key**="tXvOrlJDmLnTfiUqJ3Kuw", # OS X consumer key
oauth_nonce="CD87AE38-118A-4681-8EC4-70E8698632BE", # unique, random
oauth_signature_method="HMAC-SHA1",
oauth_timestamp="1400757042", # epoch timestamp of the request
oauth_version="1.0",
 **oauth_token**="15111995-g2Ln...", # nst021 access key for OS X
HMAC_SHA1(request_parameters, signing_key)
where signing_key is consumer_secret&access_secret
 **oauth_signature**="kzyOt09wYqj5%2FS%2FgDkfbEDxk65w%3D"

Hardcoded Tokens



TwitterAuthentication
Plugin.bundle

```
/System/Library/Accounts/Authentication/  
TwitterAuthenticationPlugin.bundle/  
Contents/MacOS/TwitterAuthenticationPlugin
```

```
uXw0slKDnLoTgiVqK3Lux # obfuscated  
tXvOrlJDmLnTfiUqJ3Kuw # clear
```

```
$ python  
>>> s = "uXw0slKDnLoTgiVqK3Lux"  
>>> print ''.join(["%c" % (ord(c) - 1) if i % 2 == 0 else c for i, c in enumerate(s)])  
tXvOrlJDmLnTfiUqJ3Kuw
```


STTwitter

- Open-source Objective-C library
- Can use SLRequest or sign OAuth requests
- Provides Obj-C method for each endpoint

<https://github.com/nst/STTwitter>

SLRequest with STTwitter

```
STTwitterAPI *t = [STTwitterAPI twitterAPIIOSWithFirstAccount];

[t verifyCredentialsWithSuccessBlock:^(NSString *username) {

    [t getUserTimelineWithScreenName:@"nst021"
        count:20
        successBlock:^(NSArray *statuses) {
            NSLog(@"-- %@", statuses);
        } errorCallback:^(NSError *error) {
            NSLog(@"-- %@", [error localizedDescription]);
        }];

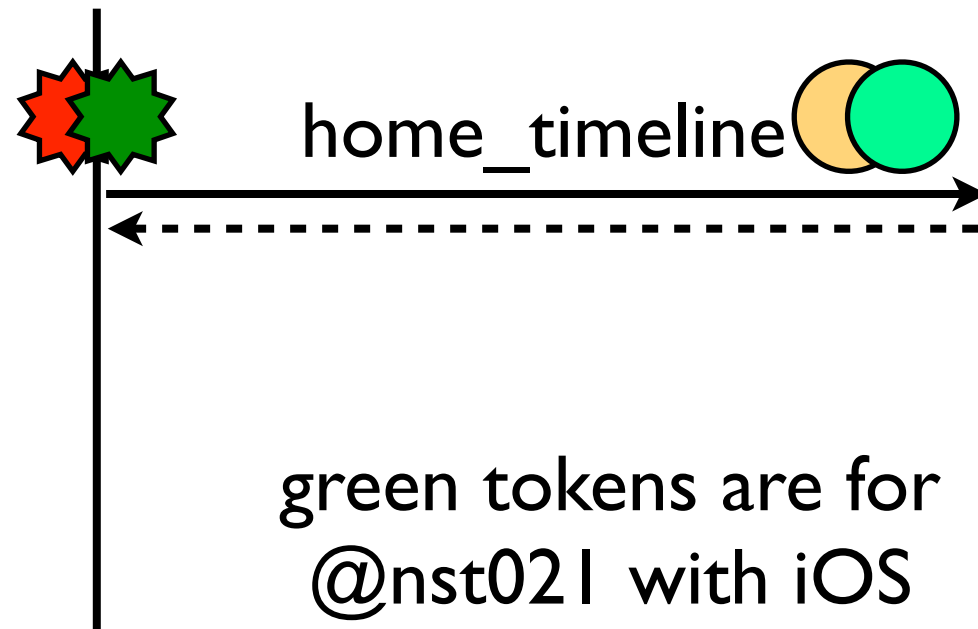
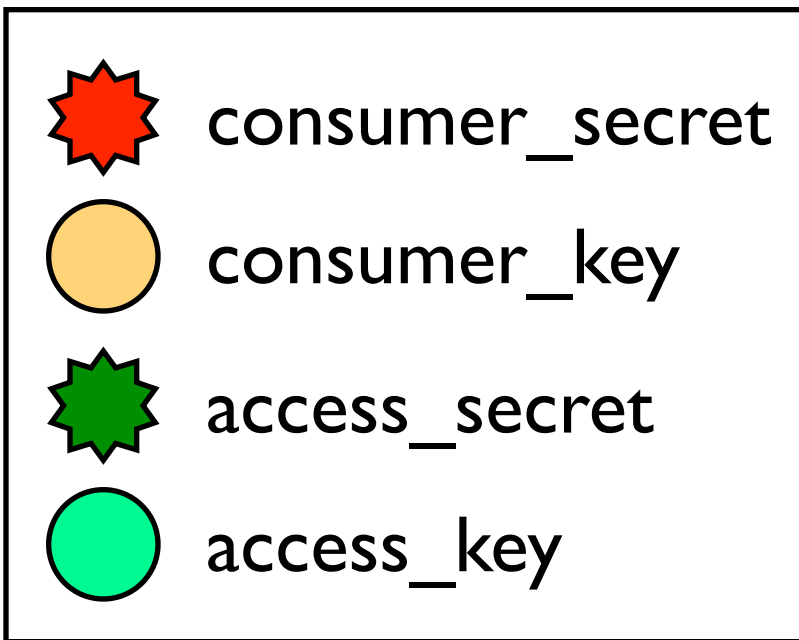
} errorCallback:^(NSError *error) {
    NSLog(@"-- %@", [error localizedDescription]);
}];
```

Notation

@nst021 / iOS

Twitter

SLRequest



Now, how do we get all these tokens in the first place?

I. Access Twitter API

I.1 Tweet Sheet

I.2 SLRequest

I.3 OAuth

I.4 XAuth

I.5 Safari

I.6 Reverse Auth

I.7 App Only

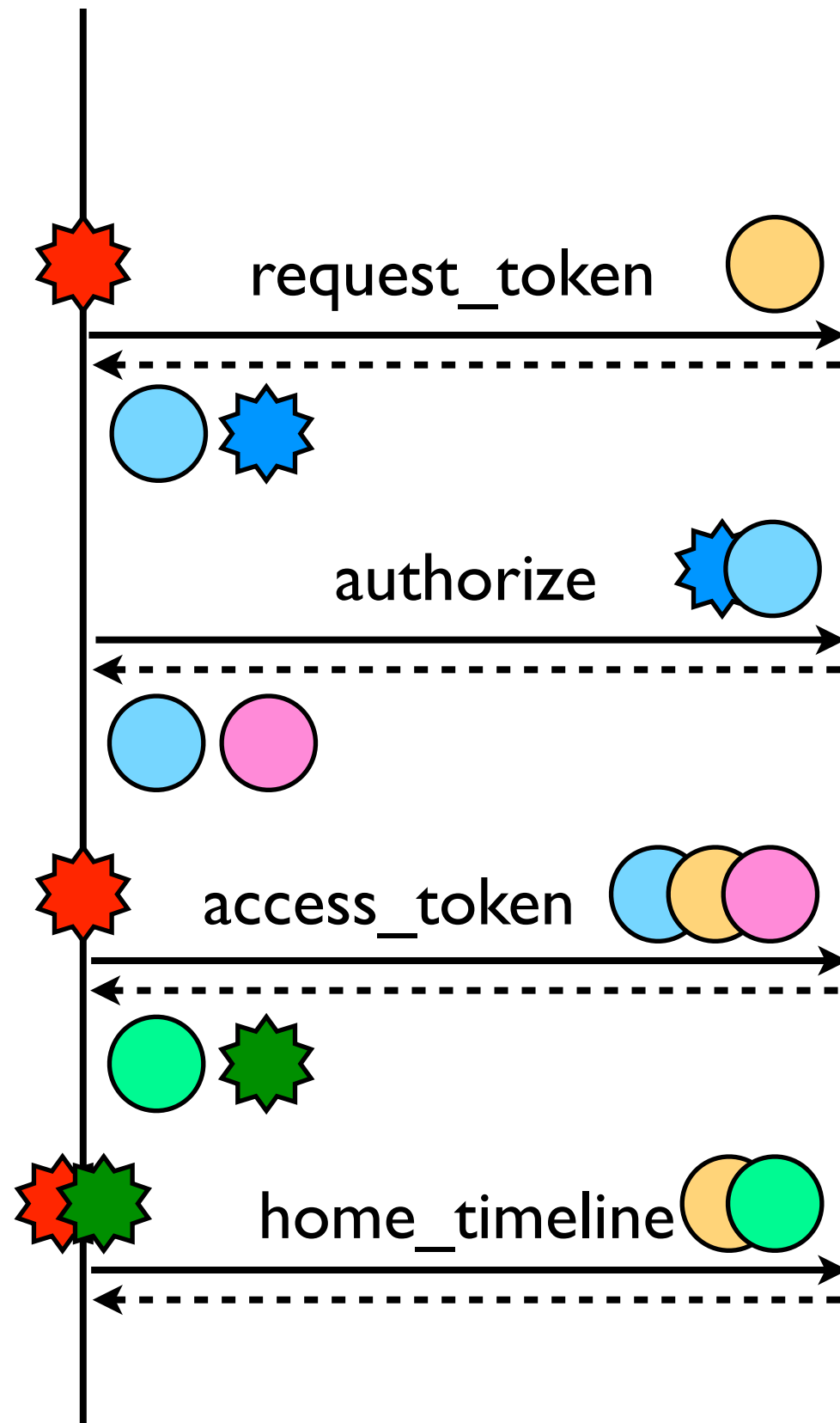
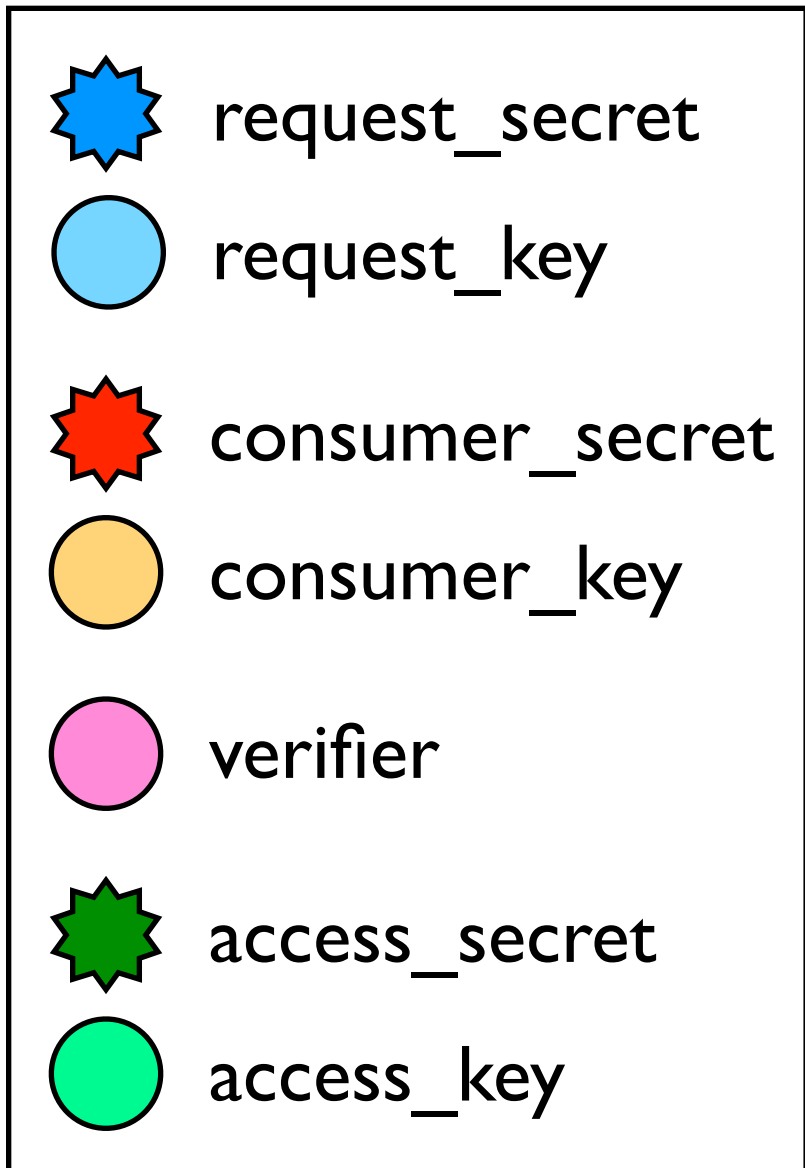
2. Twitter API

3. STTwitter

@nst021 / My.app

Twitter

Get access tokens for App/User

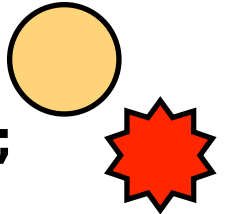


3 phases Auth.
Desktop

green tokens valid for @nst021 on My.app

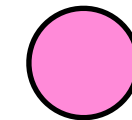
OAuth with PIN

```
STTwitterAPI *t = [STTwitterAPI twitterAPIWithOAuthConsumerKey:@"  
consumerSecret:@"];
```

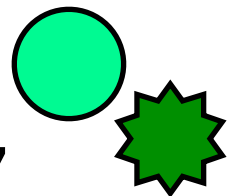


```
[t postTokenRequest:^(NSURL *url, NSString *oauthToken) {
```

```
// open url, authorize application, get the pin
```



```
[t postAccessTokenRequestWithPIN:pin  
    successBlock:^(NSString *oauthToken,  
                   NSString *oauthTokenSecret,  
                   NSString *userID,  
                   NSString *screenName) {
```



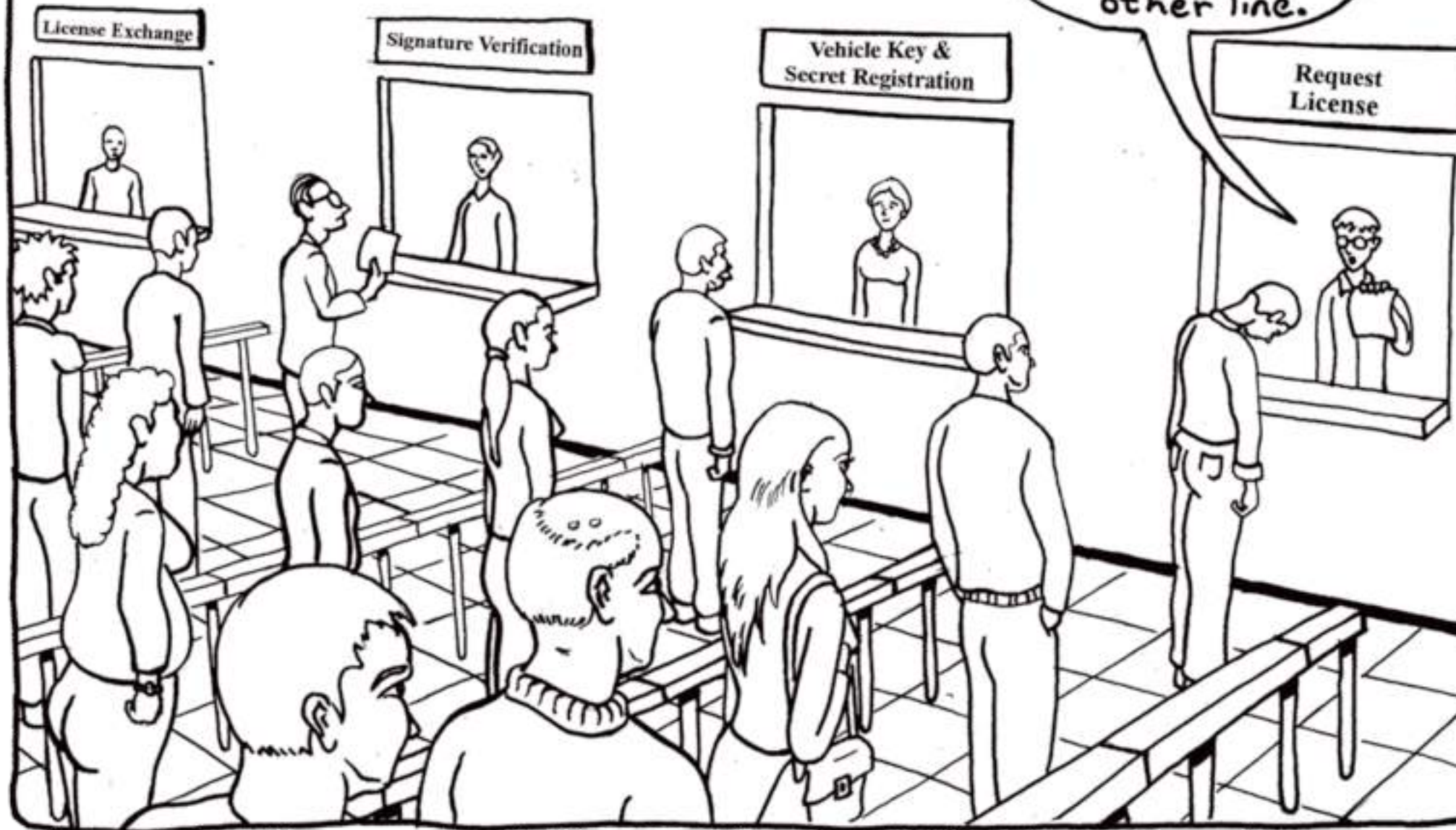
```
    } errorCallback:^(NSError *error) {
```

```
    }];
```

```
} oauthCallback:nil errorCallback:^(NSError *error) {
```

```
};
```

DEPARTMENT OF MOTOR VEHICLE
Now OAuth Enabled



<http://hueniverse.com/2007/09/oauth-isnt-always-the-solution/>

1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

1.5 Safari

1.6 Reverse Auth

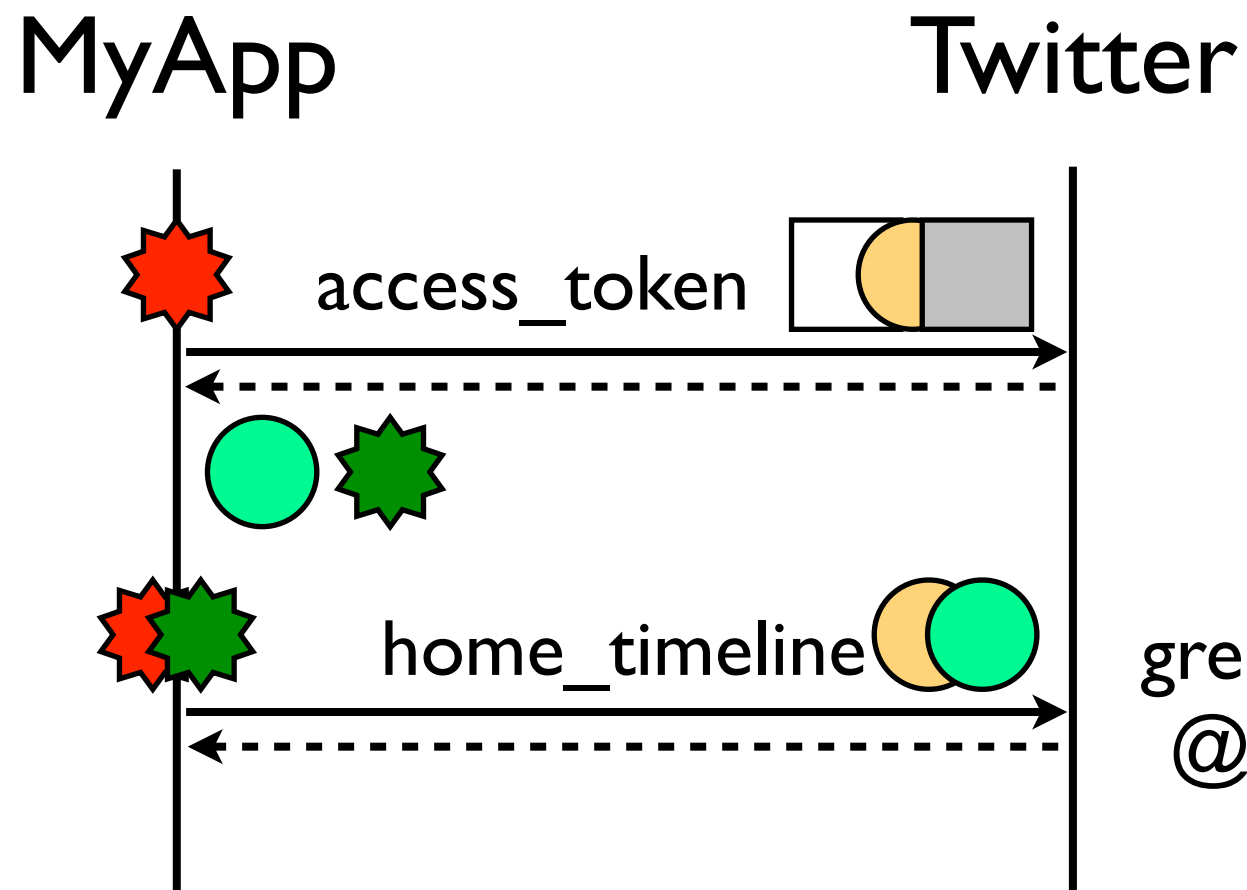
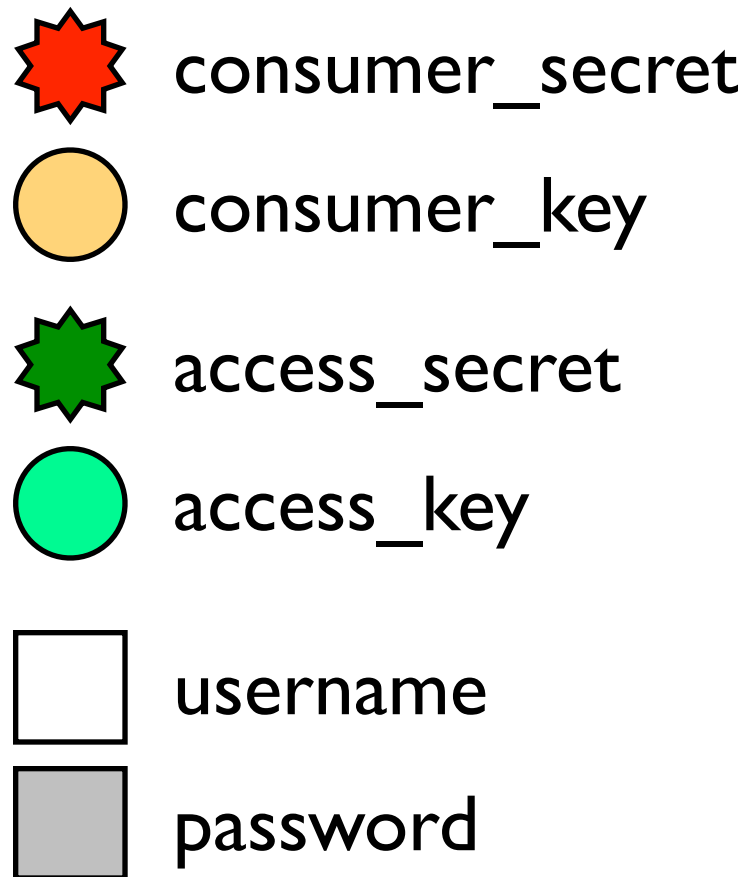
1.7 App Only

2. Twitter API

3. STTwitter

XAuth

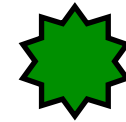
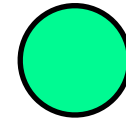
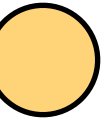
- Simplification of OAuth 3 steps auth.
- Exchange username and password for access_tokens



green tokens are for @nst021 with iOS

xAuth: 1 phase Authentication

```
STTwitterAPI *t = [STTwitterAPI twitterAPIWithOAuthConsumerKey:@"  
consumerSecret:@"  
username:@"  
password:@"];  
  
[t verifyCredentialsWithSuccessBlock:^(NSString *username) {  
    NSLog(@"-- %@", t.oauthAccessToken);  
    NSLog(@"-- %@", t.oauthAccessTokenSecret);  
    // open url, authorize application, get the pin  
}  
errorBlock:^(NSError *error) {  
    NSLog(@"-- %@", [error localizedDescription]);  
}];
```



[Home](#) → [Documentation](#) → [OAuth](#) → [Obtaining access tokens](#)

Tweet

xAuth

OAuth

View

[What links here](#)

Updated on Wed, 2013-05-22 12:40

API version 1

API version 1.1

Related Questions

- [How long do I have to wait when requesting xAuth?](#)

About xAuth

xAuth is still OAuth. You still need to master [how to send signed requests to Twitter](#).

The xAuth process will only yield read-only or read-write access tokens. Direct message read access is not provided with xAuth. If your application requires access to a user's direct messages, you will need to use the full OAuth flow.

xAuth provides a way for desktop and mobile applications to exchange a username and password for an OAuth access token. Once the access token is retrieved, xAuth-enabled developers should dispose of the login and password corresponding to the user.

xAuth access is restricted to approved applications. If your application is a desktop or mobile application that has no other recourse but to utilize xAuth, send a detailed request to <https://support.twitter.com/forms/platform>. Include the name of your application, the consumer key, the application ID (if available), and a summary of how xAuth is best-suited for your application.

xAuth requires that you use header-based OAuth authentication against an SSL access token end point, using the POST HTTP method.

xAuth also supports Twitter's [Login Verification](#) feature.

Tags

- [OAuth](#) (178)
- [Auth](#) (31)
- [xAuth](#) (11)
- [login verification](#) (1)

Using xAuth

To use xAuth, first request access by filling out [this form](#) with plenty of details about your application and why xAuth is the best choice for it. xAuth is only granted in circumstances where the standard OAuth flow is not possible.

1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

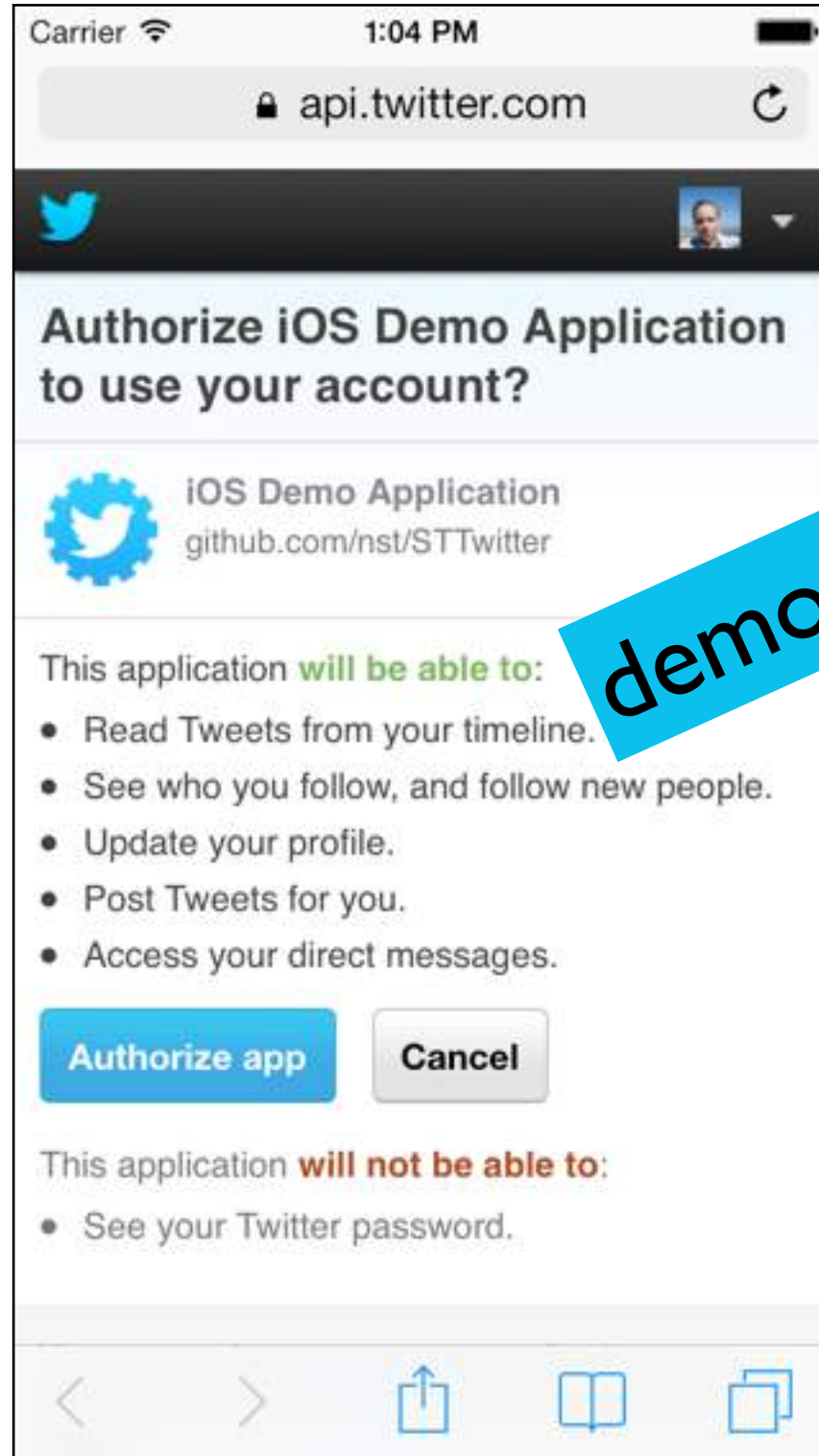
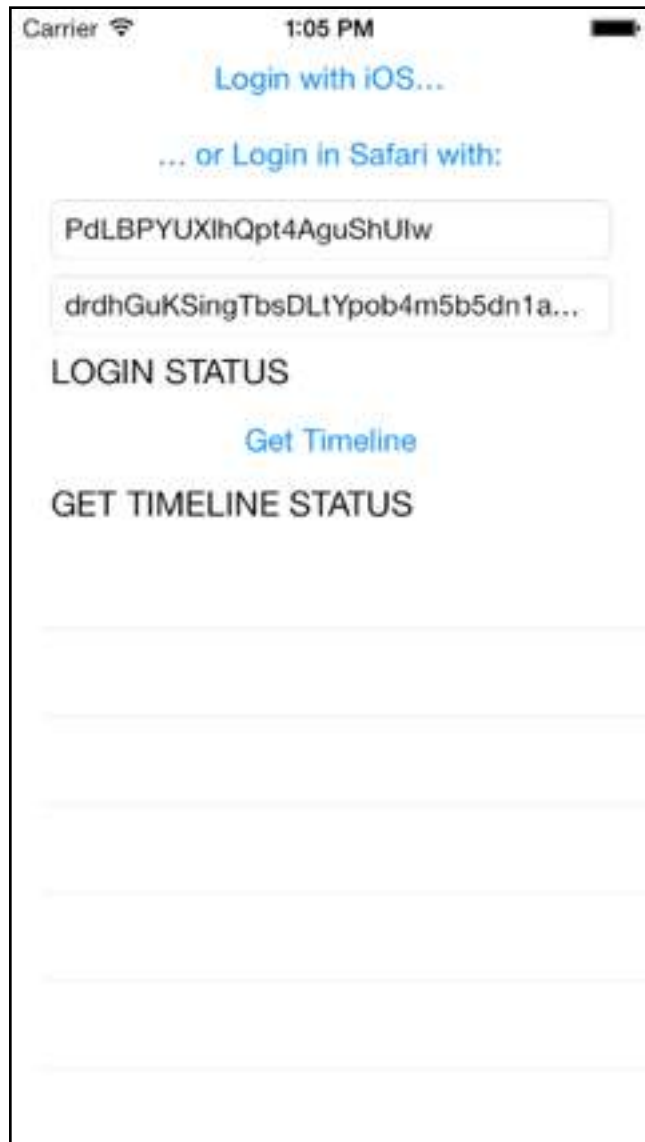
1.5 Safari

1.6 Reverse Auth

1.7 App Only

2. Twitter API

3. STTwitter



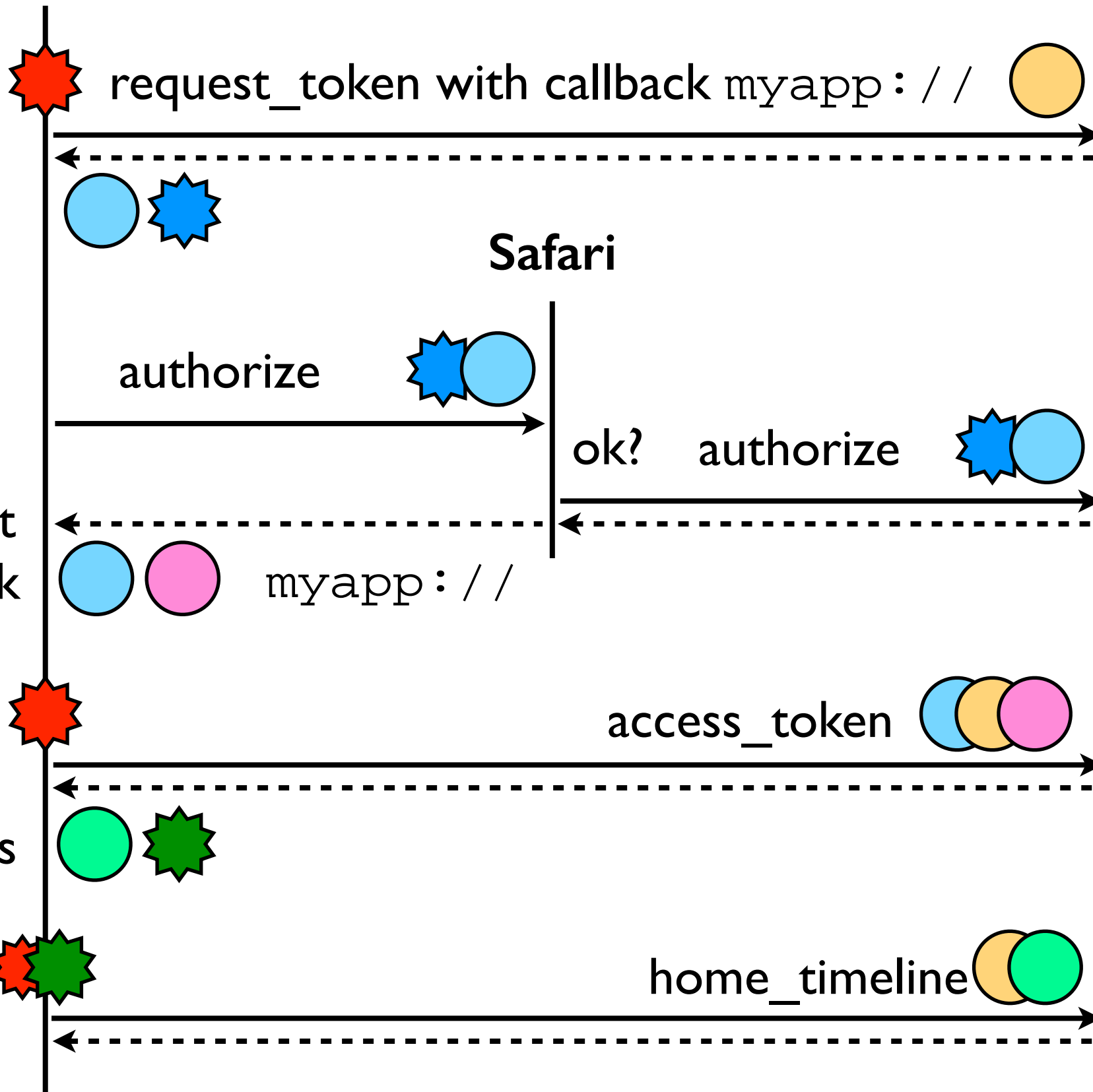
demo iOS



Benefit: trust! user doesn't give you her password.

MyApp

Twitter



The user authorizes MyApp to access his account, Twitter sends the verifier to the custom callback.

verifier is sent to the callback

access tokens

access_token

home_timeline

```
STTwitterAPI *t = [STTwitterAPI twitterAPIWithOAuthConsumerKey:@"  
consumerSecret:@"];  
  
[t postTokenRequest:^(NSURL *url, NSString *oauthToken) {  
    [[UIApplication sharedApplication] openURL:url];  
} oauthCallback:@"myapp://twitter_access_tokens/"  
    errorCallback:^(NSError *error) {  
    }  
}];
```

```
- (BOOL)application:(UIApplication *)application  
    openURL:(NSURL *)url  
    sourceApplication:(NSString *)sourceApplication  
    annotation:(id)annotation {  
    if ([[url scheme] isEqualToString:@"myapp"] == NO) return NO;  
    NSString *query = [url query];  
    // read 'oauth_verifier' in query  
    return YES;  
}
```

```
[t postAccessTokenRequestWithPIN:verifier successBlock:^(NSString *oauthToken,  
    NSString *oauthTokenSecret,  
    NSString *userID,  
    NSString *screenName) {  
    NSLog(@"-- screenName: %@", screenName);  
} errorCallback:^(NSError *error) {  
}];
```

1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

1.5 Safari

1.6 Reverse Auth

1.7 App Only

2. Twitter API

3. STTwitter

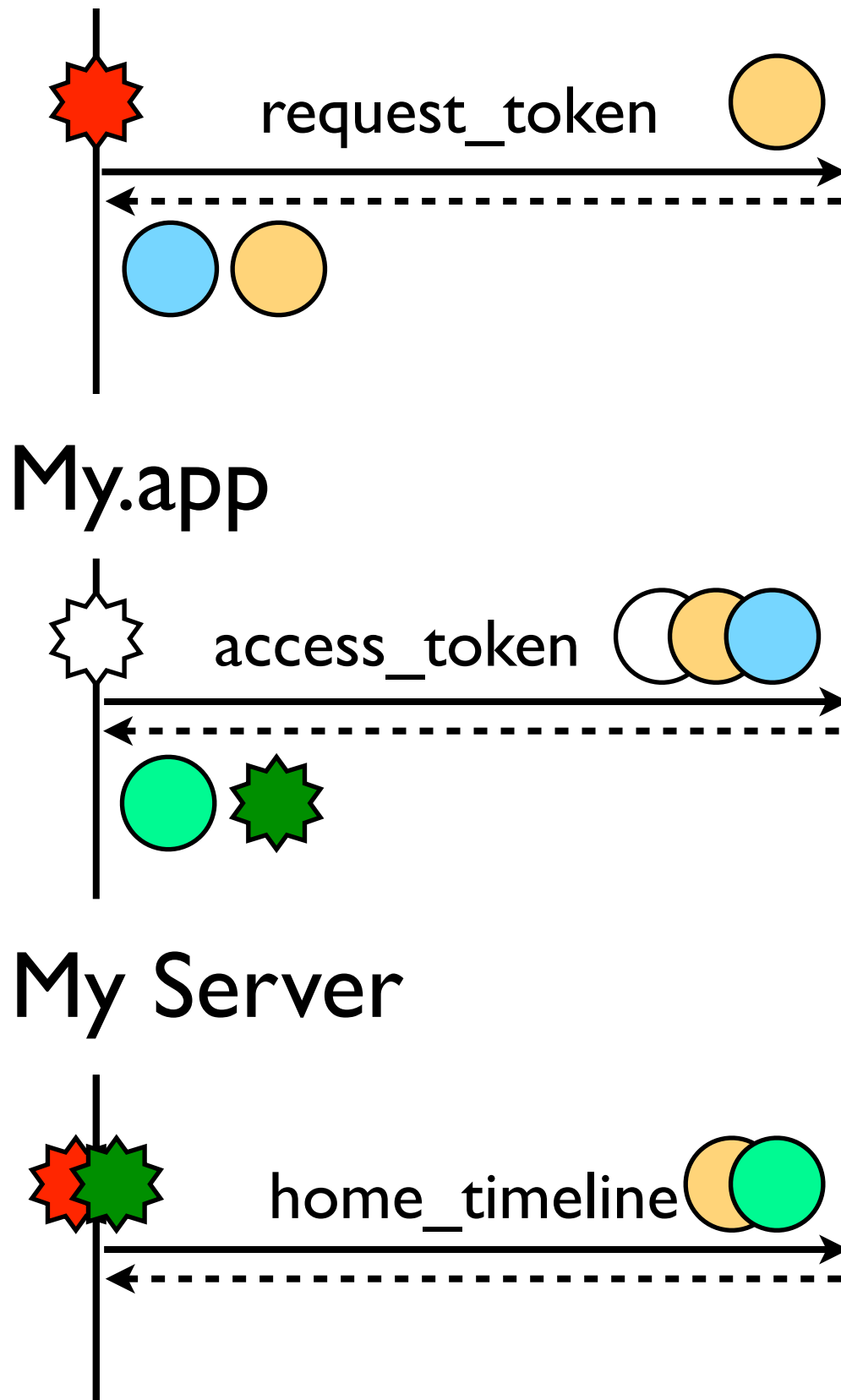
Reverse Auth

- Use case: U on A1 (iOS) => U/A1 allows A2 to get access tokens for U/ A2
- Benefits: don't ship CS, identify user with her Twitter account, access user's account from your server, access direct messages

My App or My Server

Twitter

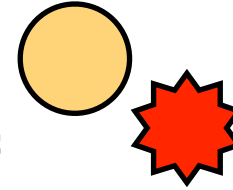
- request_key
- consumer_secret
- consumer_key
- access_secret
- access_key
- iOS_secret
- iOS_key



Reverse
Auth.

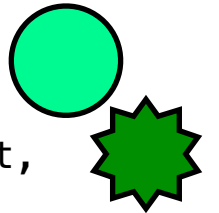
green tokens are for
@nst021 with
My.app

```
STTwitterAPI *twitter = [STTwitterAPI twitterAPIWithOAuthConsumerKey:@"  
consumerSecret:@"];
```



```
[twitter postReverseOAuthTokenRequest:^(NSString *authenticationHeader) {  
    STTwitterAPI *twitterAPIOS = [STTwitterAPI twitterAPIOSWithFirstAccount];  
    [twitterAPIOS verifyCredentialsWithSuccessBlock:^(NSString *username) {
```

```
        [twitterAPIOS postReverseAuthAccessTokenWithAuthenticationHeader:authenticationHeader  
                    successBlock:^(NSString *oAuthToken,  
                                    NSString *oAuthTokenSecret,  
                                    NSString *userID,  
                                    NSString *screenName) {
```



```
                // use the tokens...
```

```
        } errorCallback:^(NSError *error) {  
            // ...  
        }];
```

```
    } errorCallback:^(NSError *error) {  
        // ...  
    }];
```

```
} errorCallback:^(NSError *error) {  
    // ...  
}];
```

1. Access Twitter API

1.1 Tweet Sheet

1.2 SLRequest

1.3 OAuth

1.4 XAuth

1.5 Safari

1.6 Reverse Auth

1.7 App Only

2. Twitter API

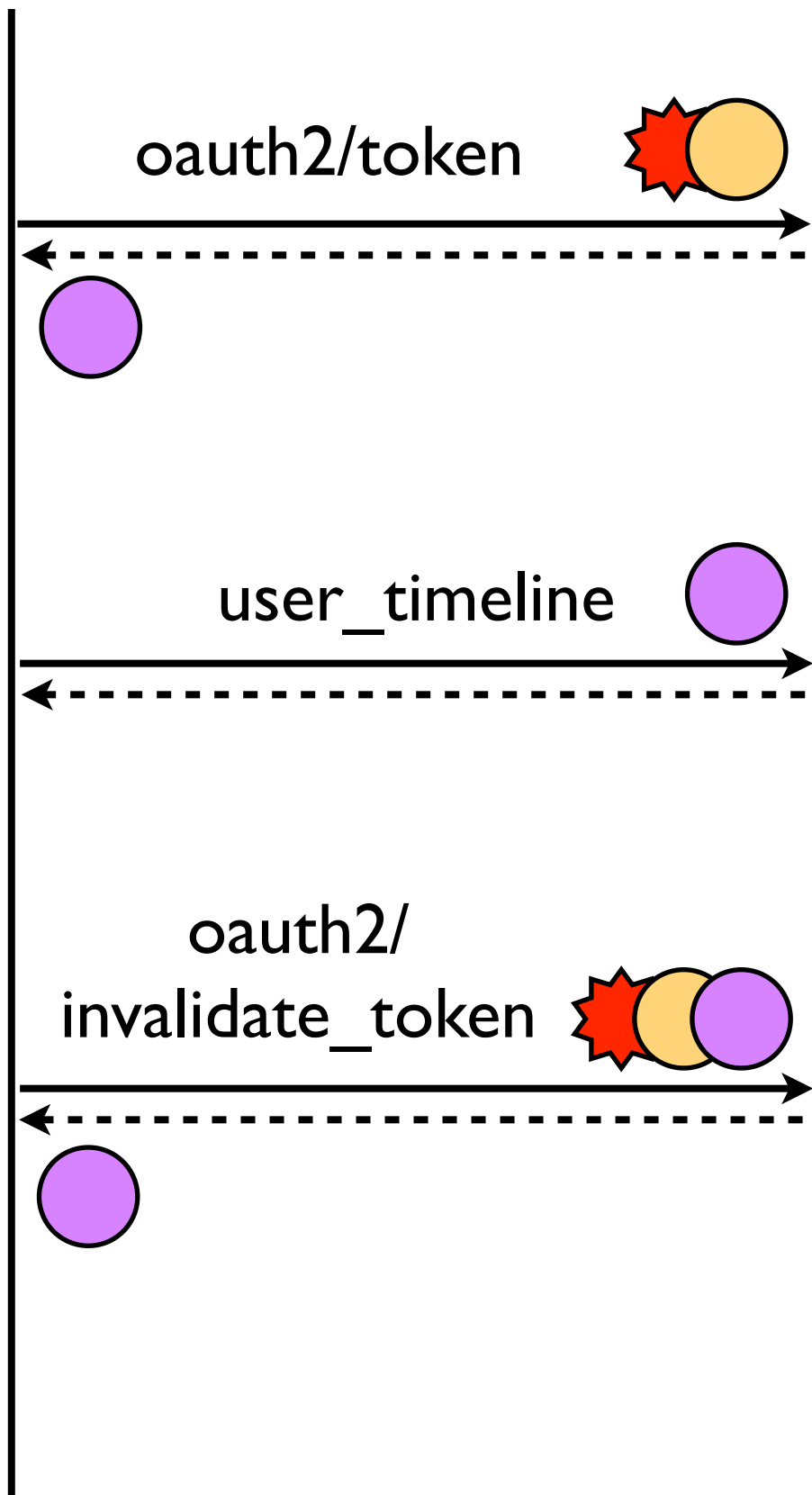
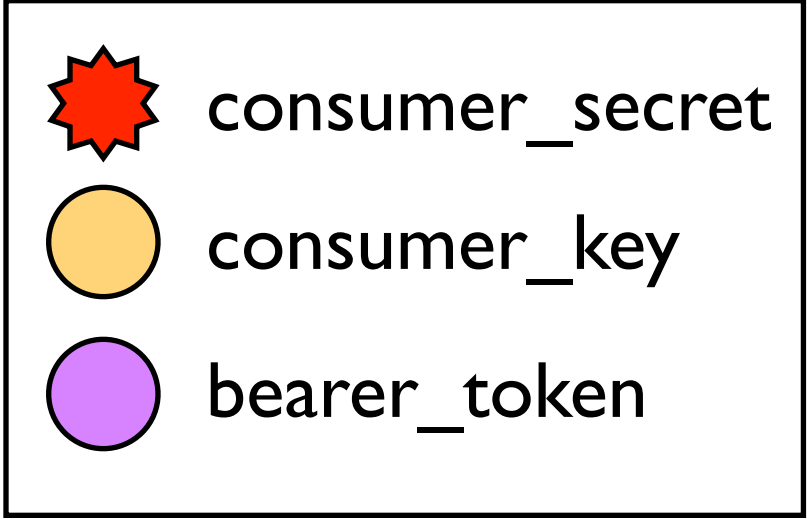
3. STTwitter

App Only

- **Use case: no user's context**
- **Eg. display your company's timeline in your iOS application**
- **Benefit: you can use Twitter API without users having a Twitter account**

iOS

Twitter



App. Only
Authentication

violet token is for iOS

demo OS X

App Only

```
STTwitterAPI *t = [STTwitterAPI twitterAPIAppOnlyWithConsumerKey:@"  
consumerSecret:@"];
```

```
[_twitter verifyCredentialsWithSuccessBlock:^(NSString *bearerToken) {
```

```
    // use API without user's context
```

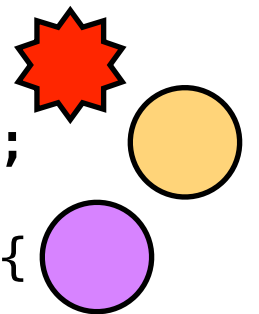
```
    // eg. display timeline for @BarackObama
```

```
    // but cannot post or favorite a tweet because no user context
```

```
} errorCallback:^(NSError *error) {
```

```
    //
```

```
}];
```



1. Access Twitter API

2. Twitter API

3. STTwitter

Twitter API

<https://dev.twitter.com/docs/api/1.1>

Timeline,
Retweet,
Search,
Direct Messages
...

Follow / Unfollow
Favorites
Lists
Trends
...

API Changes

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

<https://dev.twitter.com/docs/recent>

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

<https://twitter.com/twitterapi>

Rate Limits in Headers

```
{  
  "x-rate-limit-limit" = 15;  
  "x-rate-limit-remaining" = 14;  
  "x-rate-limit-reset" = 1381054457;  
}
```

Undocumented API?

```
$ strings Twitter
```

```
...
```

```
activity/about_me.json
```

```
activity/by_friends.json
```

```
conversation/show.json
```

```
discover/highlight.json
```

```
discover/universal.json
```

```
statuses/:id/activity/summary.json
```

```
statuses/media_timeline.json
```

```
statuses/mentions_timeline.json
```

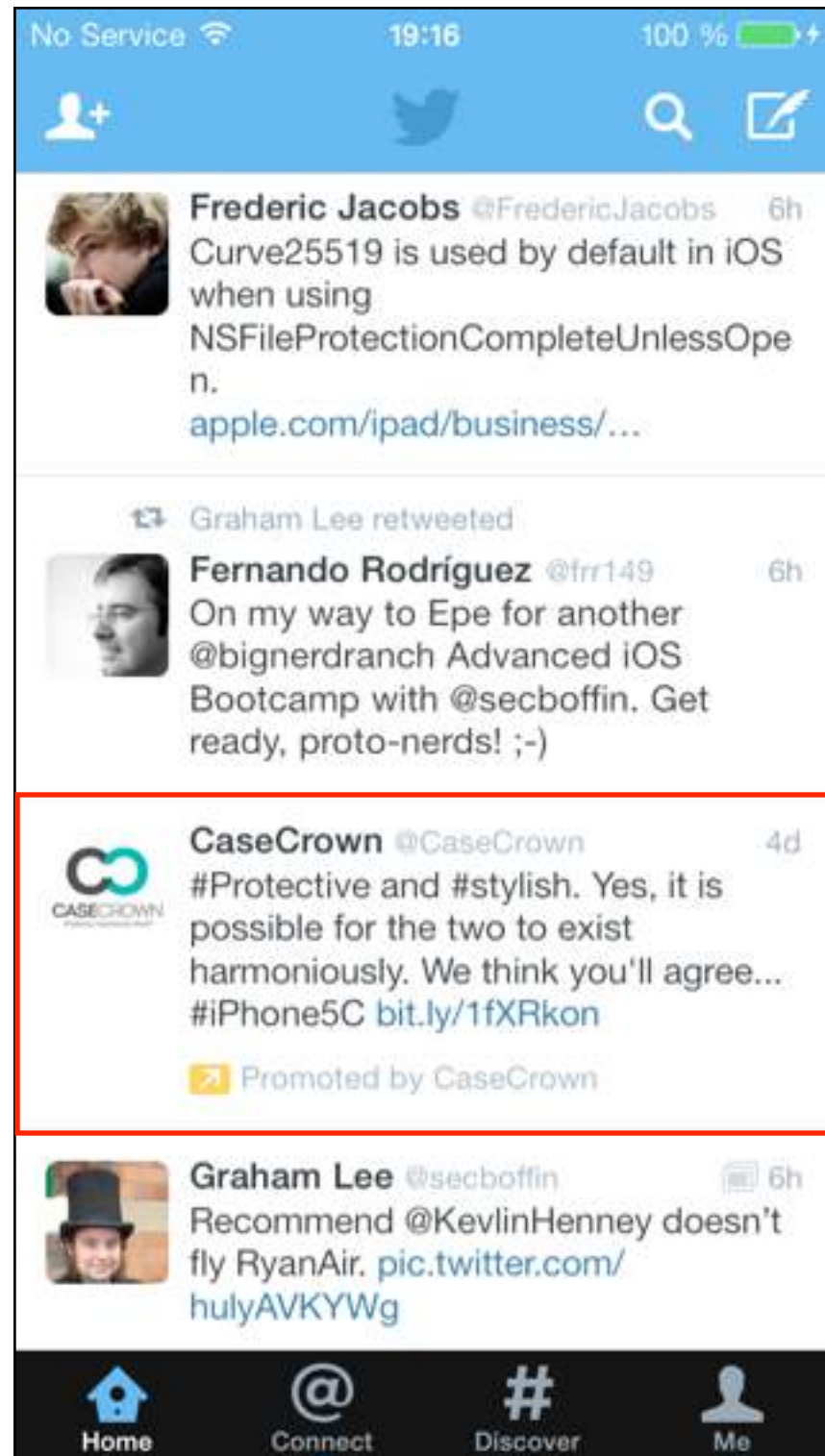
```
timeline/home.json
```

```
trends/available.json
```

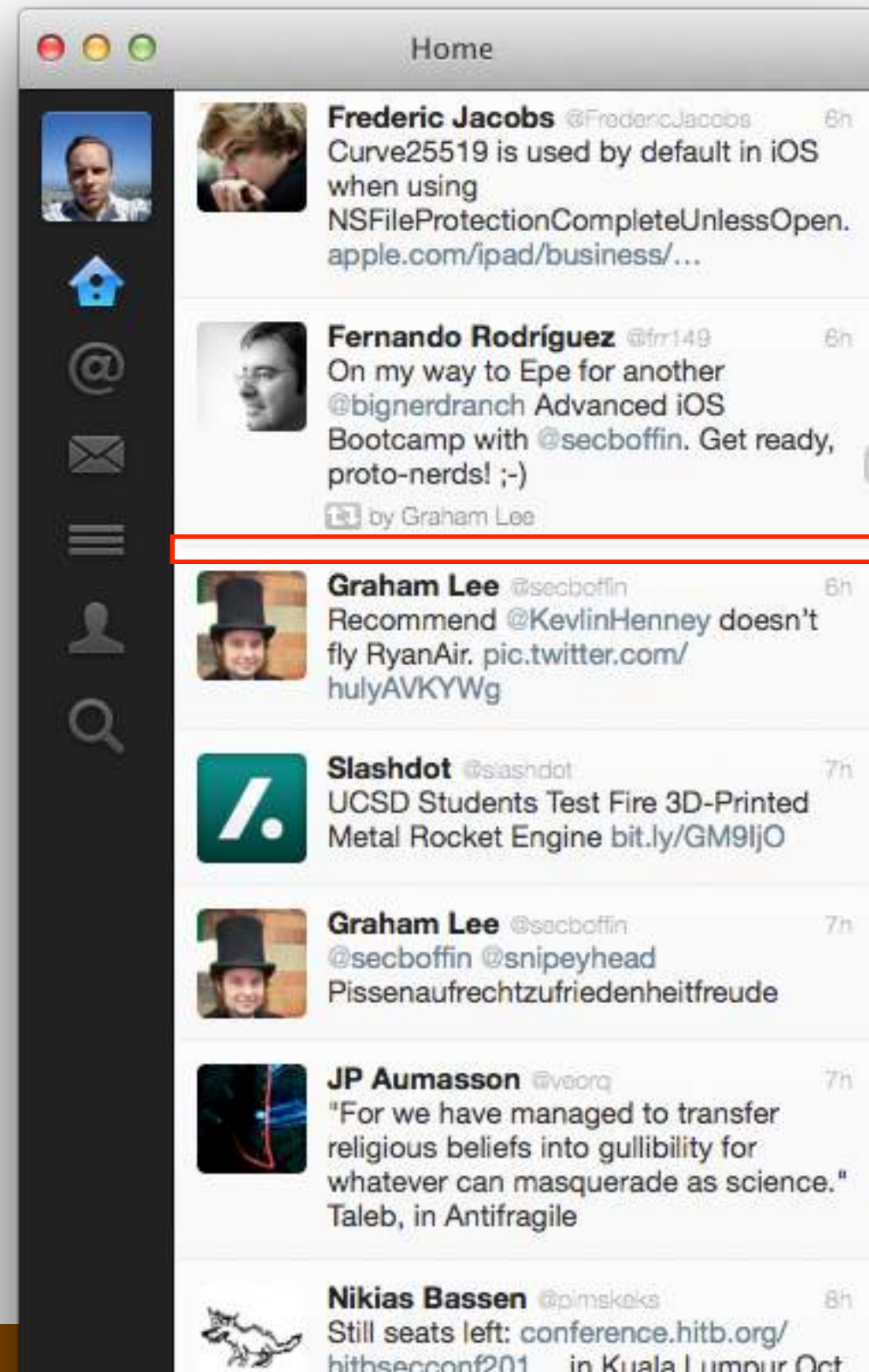
```
users/recommendations.json
```

```
...
```

Promoted Contents



Ad



No Ad

Removing Certificate Pinning

```
-[ABHTTPRequest connection:willSendRequestForAuthenticationChallenge:]
```

```
...
```

```
0x00260cd4 45F2EA50    movw    r0, #0x55ea
0x00260cd8 3246        mov     r2, r6
0x00260cda C0F23900    movt   r0, #0x39
0x00260cde 7844        add     r0, pc      ; 0x5f62cc
0x00260ce0 0168        ldr    r1, [r0]    ; @selector(_isPinnedCertificateChain:)
0x00260ce2 2046        mov     r0, r4
0x00260ce4 B0F1DEEC    blx    imp___picsymbolstub4__objc_msgSend
```

```
; BOOL isPinned = [self _isPinnedCertificateChain:object];
```

```
0x00260ce8 0446        mov     r4, r0
0x00260cea 2846        mov     r0, r5
0x00260cec B0F1FAEC    blx    imp___picsymbolstub4__objc_release
```

```
- 0x00260cf0 14F0FF0F    tst.w   r4, #0xff    ; Z = (r4 & 0xff) == 0 ; Z = (r4 == 0)
+ 0x00260cf0 90E0FF0F    b       0x260e14    ; jump to happy path (FF0F is unused)
0x00260cf4 40F08E80    bne.w  0x260e14 => ; never reached
```

```
...
```


My 2 Bytes

25CC20	A821 7844 C0F2 3801 7944 0068 D1F8 00A0	25CC20	A821 7844 C0F2 3801 7944 0068 D1F8 00A0
25CC30	0268 2046 5146 B0F1 36ED 8046 2046 B0F1	25CC30	0268 2046 5146 B0F1 36ED 8046 2046 B0F1
25CC40	52ED 2846 B0F1 4EED 18F0 FF0F 00F0 8980	25CC40	52ED 2846 B0F1 4EED 18F0 FF0F 00F0 8980
25CC50	5846 3146 B0F1 26ED 3F46 B0F1 64ED 0446	25CC50	5846 3146 B0F1 26ED 3F46 B0F1 64ED 0446
25CC60	44F6 8440 C0F2 3900 7844 0168 2046 B0F1	25CC60	44F6 8440 C0F2 3900 7844 0168 2046 B0F1
25CC70	1AED 0646 2046 B0F1 36ED 45F2 3A60 3246	25CC70	1AED 0646 2046 B0F1 36ED 45F2 3A60 3246
25CC80	C0F2 3900 019C 7844 0168 2046 B0F1 0AED	25CC80	C0F2 3900 019C 7844 0168 2046 B0F1 0AED
25CC90	10F0 FF0F 78D0 4DF2 F660 C0F2 3B00 4FF2	25CC90	10F0 FF0F 78D0 4DF2 F660 C0F2 3B00 4FF2
25CCA0	0441 7844 C0F2 3801 7944 0068 0968 2058	25CCA0	0441 7844 C0F2 3801 7944 0068 0968 2058
25CCB0	B0F1 F8EC 3F46 B0F1 36ED 4EF6 9A22 5146	25CCB0	B0F1 F8EC 3F46 B0F1 36ED 4EF6 9A22 5146
25CCC0	C0F2 3A02 0546 7A44 B0F1 ECEC 10F0 FF0F	25CCC0	C0F2 3A02 0546 7A44 B0F1 ECEC 10F0 FF0F
25CCD0	00F0 9D80 45F2 EA50 3246 C0F2 3900 7844	25CCD0	00F0 9D80 45F2 EA50 3246 C0F2 3900 7844
25CCE0	0168 2046 B0F1 DEEC 0446 2846 B0F1 FAEC	25CCE0	0168 2046 B0F1 DEEC 0446 2846 B0F1 FAEC
25CCF0	14F0 FF0F 40F0 8E80 4EF6 2450 C0F2 3800	25CCF0	90E0 FF0F 40F0 8E80 4EF6 2450 C0F2 3800
25CD00	48F6 AA22 C0F2 3902 7844 7A44 0168 1068	25CD00	48F6 AA22 C0F2 3902 7844 7A44 0168 1068
25CD10	4EF6 5822 C0F2 3A02 7A44 B0F1 C4EC 3F46	25CD10	4EF6 5822 C0F2 3A02 7A44 B0F1 C4EC 3F46
25CD20	B0F1 00ED 46F6 9571 0546 C0F2 2201 47F2	25CD20	B0F1 00ED 46F6 9571 0546 C0F2 2201 47F2
25CD30	A413 C0F2 2203 7944 7B44 0120 4FF4 1972	25CD30	A413 C0F2 2203 7944 7B44 0120 4FF4 1972
25CD40	0095 FCF7 73F9 2846 B0F1 CCEC 4FF2 0870	25CD40	0095 FCF7 73F9 2846 B0F1 CCEC 4FF2 0870
25CD50	C0F2 3800 7844 0168 5846 B0F1 A4EC 3F46	25CD50	C0F2 3800 7844 0168 5846 B0F1 A4EC 3F46
25CD60	46E0 4FF2 F260 C0F2 3800 7844 0168 5846	25CD60	46E0 4FF2 F260 C0F2 3800 7844 0168 5846
25CD70	B0F1 98EC 3F46 B0F1 D6EC 0446 45F2 5050	25CD70	B0F1 98EC 3F46 B0F1 D6EC 0446 45F2 5050
25CD80	C0F2 3900 7844 3BE0 4EF6 9440 C0F2 3800	25CD80	C0F2 3900 7844 3BE0 4EF6 9440 C0F2 3800
25CD90	48F6 1A22 C0F2 3902 7844 7A44 0168 1068	25CD90	48F6 1A22 C0F2 3902 7844 7A44 0168 1068

1: Replace 2 bytes at offset 0x25ccf0 with 2 bytes

Promoted Contents Secret

```
nst — Python — 93x24
>> GET https://api.twitter.com/1.1/statuses/home_timeline.json?cards_platform=iPad-3&contributor_details=1&count=100&earned=true&include_cards=1&include_entities=1&include_my_retweet=1&include_user_entities=true&lat=46.6[REDACTED]&long=6.7[REDACTED]&pc=true&send_error_codes=1&since_id=388405594975387649
  ← 200 application/json 28B 157.7kB/s
POST https://twitter.com/scribe
  ← 200 text/html [no content] 99.68kB/s
POST https://twitter.com/scribe
  ← 200 text/html [no content] 145.42kB/s
GET https://api.twitter.com/1.1/activity/about_me.json?cards_platform=iPad-3&contributor_details=1&count=50&include_cards=1&include_entities=1&include_my_retweet=1&include_user_entities=true&latest_results=true&model_version=1&send_error_codes=1&since_id=1381437936000
  ← 200 application/json 28B 228.61kB/s
GET https://api.twitter.com/1.1/activity/by_friends.json?cards_platform=iPad-3&contributor_details=1&count=50&include_cards=1&include_entities=1&include_my_retweet=1&include_user_entities=true&latest_results=true&send_error_codes=1&since_id=1381438178081
  ← 200 application/json 28B 233.77kB/s
GET https://api.twitter.com/1.1/direct_messages.json?count=100&include_entities=1&include_user_entities=true&send_error_codes=1&since_id=387736111004385280
  ← 200 application/json 28B 108.14kB/s
GET https://api.twitter.com/1.1/direct_messages/sent.json?count=100&include_entities=1&inc
[44/98]                                     ? :help [*:8080]
```

&pc=true



1. Using Twitter API

2. Twitter API

3. STTwitter

Timelines

- `M -getStatusesMentionTimelineWithCount:sinceID:maxID:trimUser:contributorDetails:includeEntities:successBlock:errorBlock:`
- `M -getMentionsTimelineSinceID:count:successBlock:errorBlock:`
- `M -getStatusesUserTimelineForUserID:screenName:sinceID:count:maxID:trimUser:excludeReplies:contributorDetails:includeRetweets:successBlock:errorBlock:`
- `M -getUserTimelineWithScreenName:sinceID:maxID:count:successBlock:errorBlock:`
- `M -getUserTimelineWithScreenName:count:successBlock:errorBlock:`
- `M -getUserTimelineWithScreenName:successBlock:errorBlock:`
- `M -getStatusesHomeTimelineWithCount:sinceID:maxID:trimUser:excludeReplies:contributorDetails:includeEntities:successBlock:errorBlock:`
- `M -getHomeTimelineSinceID:count:successBlock:errorBlock:`
- `M -getStatusesRetweetsOfMeWithCount:sinceID:maxID:trimUser:includeEntities:includeUserEntities:successBlock:errorBlock:`
- `M -getStatusesRetweetsOfMeWithSuccessBlock:errorBlock:`

Tweets

- `M -getStatusesRetweetsForID:count:trimUser:successBlock:errorBlock:`
- `M -getStatusesShowID:trimUser:includeMyRetweet:includeEntities:successBlock:errorBlock:`
- `M -postStatusesDestroy:trimUser:successBlock:errorBlock:`
- `M -postStatusUpdate:inReplyToStatusID:latitude:longitude:placeID:displayCoordinates:trimUser:successBlock:errorBlock:`
- `M -postStatusRetweetWithID:trimUser:successBlock:errorBlock:`
- `M -postStatusRetweetWithID:successBlock:errorBlock:`
- `M -postStatusUpdate:mediaDataArray:possiblySensitive:inReplyToStatusID:latitude:longitude:placeID:displayCoordinates:uploadProgressBlock:successBlock:errorBlock:`
- `M -postStatusUpdate:inReplyToStatusID:mediaURL:placeID:latitude:longitude:uploadProgressBlock:successBlock:errorBlock:`
- `M -getStatusesOEmbedForStatusID:urlString:maxWidth:hideMedia:hideThread:omitScript:align:related:lang:successBlock:errorBlock:`
- `M -getStatusesRetweetersIDsForStatusID:cursor:successBlock:errorBlock:`

Search

- `M -getSearchTweetsWithQuery:geocode:lang:locale:resultType:count:until:sinceID:maxID:includeEntities:callback:successBlock:errorBlock:`
- `M -getSearchTweetsWithQuery:successBlock:errorBlock:`

Streaming

- `M -postStatusesFilterUserIDs:keywordsToTrack:locationBoundingBoxes:delimited:stallWarnings:progressBlock:stallWarningBlock:errorBlock:`
- `M -postStatusesFilterKeyword:progressBlock:errorBlock:`
- `M -getStatusesSampleDelimited:stallWarnings:progressBlock:stallWarningBlock:errorBlock:`
- `M -getStatusesFirehoseWithCount:delimited:stallWarnings:progressBlock:stallWarningBlock:errorBlock:`
- `M -getUserStreamDelimited:stallWarnings:includeMessagesFromFollowedAccounts:includeReplies:keywordsToTrack:locationBoundingBoxes:progressBlock:stallWarningBlock:errorBlock:`
- `M -getSiteStreamForUserIDs:delimited:stallWarnings:restrictToUserMessages:includeReplies:progressBlock:stallWarningBlock:errorBlock:`

Direct Messages

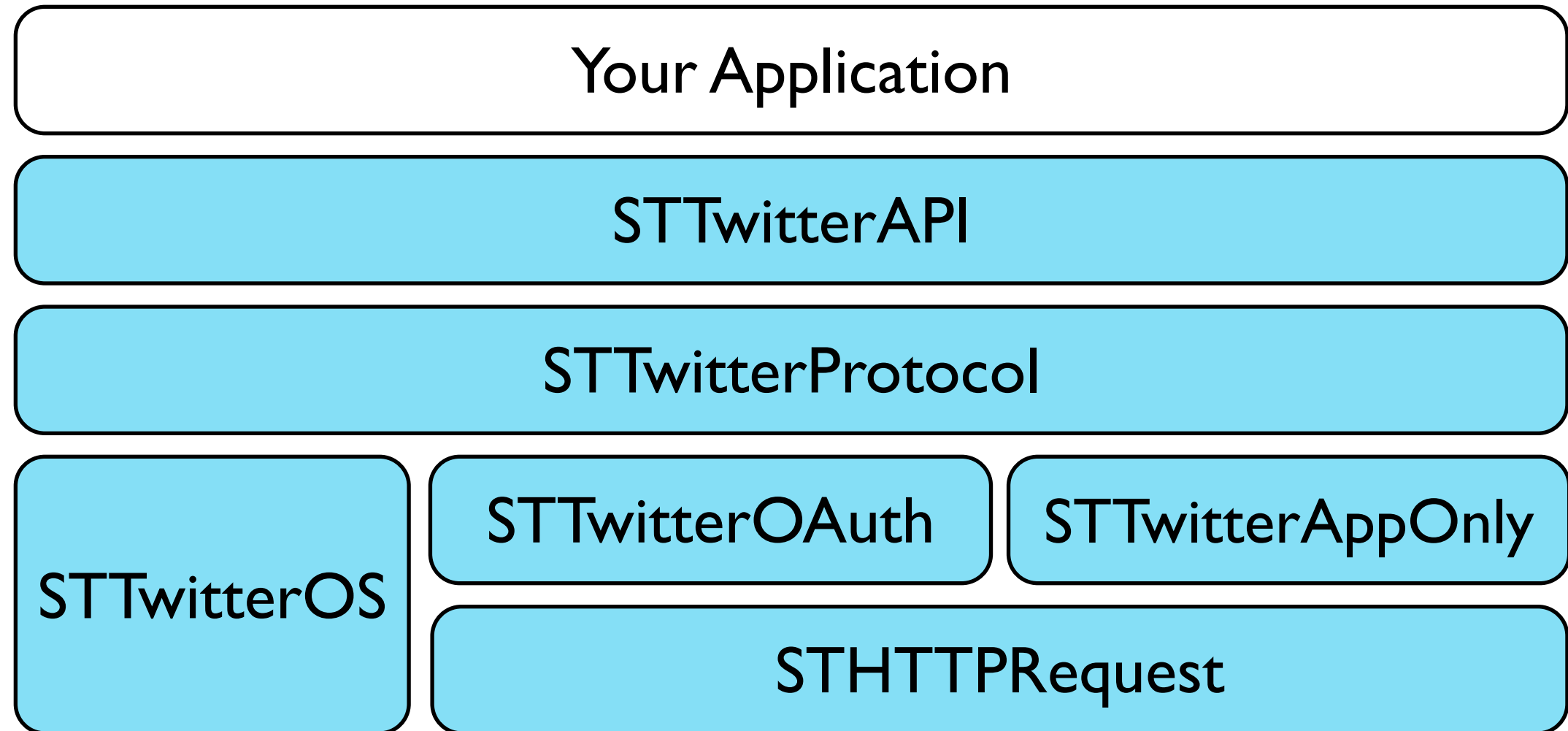
- `M -getDirectMessagesSinceID:maxID:count:includeEntities:skipStatus:successBlock:errorBlock:`
- `M -getDirectMessagesSinceID:count:successBlock:errorBlock:`
- `M -getDirectMessagesSinceID:maxID:count:page:includeEntities:successBlock:errorBlock:`
- `M -getDirectMessagesShowWithID:successBlock:errorBlock:`
- `M -postDestroyDirectMessageWithID:includeEntities:successBlock:errorBlock:`
- `M -postDirectMessage:to:successBlock:errorBlock:`

Friends & Followers

UNDOCUMENTED APIs

- `M` `-_getActivityAboutMeSinceID:count:includeCards:modelVersion:sendErrorCodes:contributorDetails:includeEntities:includeMyRetweet:successBlock:errorBlock:`
- `M` `-_getActivityByFriendsSinceID:count:contributorDetails:includeCards:includeEntities:includeMyRetweets:includeUserEntites:latestResults:sendErrorCodes:successBlock:errorBlock:`
- `M` `-_getStatusesActivitySummaryForStatusID:successBlock:errorBlock:`
- `M` `-_getConversationShowForStatusID:successBlock:errorBlock:`
- `M` `-_getDiscoverHighlightWithSuccessBlock:errorBlock:`
- `M` `-_getDiscoverUniversalWithSuccessBlock:errorBlock:`
- `M` `-_getMediaTimelineWithSuccessBlock:errorBlock:`
- `M` `-_getUsersRecommendationsWithSuccessBlock:errorBlock:`
- `M` `-_getTimelineHomeWithSuccessBlock:errorBlock:`
- `M` `-_getStatusesMentionsTimelineWithCount:contributorsDetails:includeEntities:includeMyRetweet:successBlock:errorBlock:`
- `M` `-_getTrendsAvailableWithSuccessBlock:errorBlock:`
- `M` `-_postUsersReportSpamForTweetID:reportAs:blockUser:successBlock:errorBlock:`
- `M` `-_postAccountGenerateWithADC:discoverableByEmail:email:geoEnabled:language:name:password:screenName:sendErrorCode:timeZone:successBlock:errorBlock:`
- `M` `-_getSearchTypeaheadQuery:resultType:sendErrorCodes:successBlock:errorBlock:`

Architecture




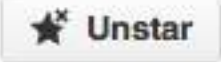
Account.framework
Social.framework

STHTTPRequest


```
STHTTPRequest *r = [STHTTPRequest requestWithURLString:@"http://twitter.com"];  
  
r.completionBlock = ^(NSDictionary *headers, NSString *body) {  
    // ...  
};  
  
r.errorBlock = ^(NSError *error) {  
    // ...  
};  
  
[r startAsynchronous];
```

Trending in Objective-C repositories last week

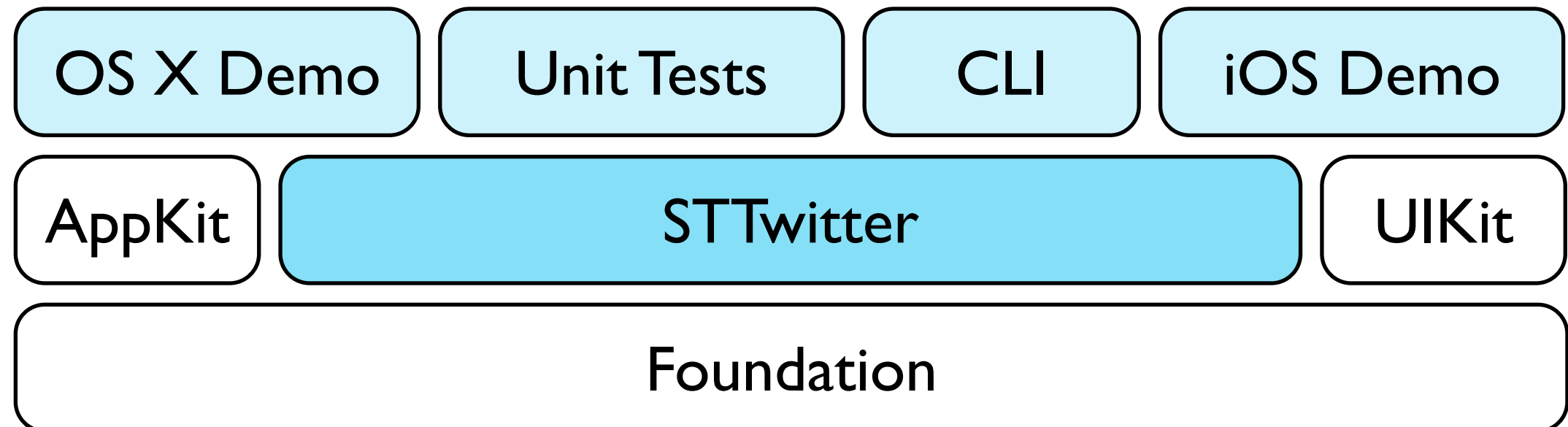
Repositories Developers Trending: today ▾ All languages
Unknown languages
C
C++
Java
JavaScript
Objective-C

1  **nst/STHTTPRequest** Objective-C  ★ Unstar

A NSURLConnection wrapper for humans

★ 270 📄 7 built by 

Subprojects



CLI demos

```
nst:Debug nst$ ./clitter
-----
Sun Oct 20 18:22:44 +0000 2013 @0xcd favored:
[F 1] [R 0] @nuthatch      Hmm. I was using SimPolders but now I've discovered Simulator Folders from @john_nye
-----
Sun Oct 20 17:51:02 +0000 2013 @FredericJacobs favored:
[F 1] [R 0] @noneinhere          you know how kids are always complaining about school around here? Kids in #Syria love school and those who don't go to school would die to.
[F 2] [R 0] @JZdziarski          If you haven't pulled apart router firmware bundles before, binwalk is a great tool for identifying different components in a .bin.
[F 2] [R 4] @noneinhere          "Those who make peaceful revolution impossible will make violent revolution inevitable" #FreeSyria #Syria #act2EndAssadsWar
[F 1] [R 1] @lifewinning          Serious question: are people who don't work at NGOs actually going to the #stopwatchingus rally?
[F 2] [R 0] @whispersystems      @joshualund The only secrets are in the mind of @corbett. There is a secret RedPhone repo though.
[F 1] [R 0] @pbililou            @FredericJacobs Yup. Though what I especially like about @AbiSaworth 's article is how it distanciates itself from weird japanese sex stuff.
[F 3] [R 13] @AmalHanano           Many people in #Syria this year are not slaughtering lambs for Eid. Meat has become a dream commodity to millions.
[F 1] [R 2] @briannoguchi         "In theory, theory and practice are the same. In practice, they are not." - Albert Einstein
[F 4] [R 3] @Bedouinocracy       Palestinian prisoners freed from zionist torture chambers and syrian prisoners freed from the Assad gulags. Thats when I'll celebrate.
[F 4] [R 3] @Ugariti_Homei        Until now I haven't lost hope that I'll see Homs again at some point!
[F 52] [R 129] @iamdeveloper  What's long and hard?

Objective-C method names.
[F 2] [R 1] @TelecomixSyria       blip blip, reboot sucessful! #system
[F 4] [R 0] @Dymaxion             (Also, hello Switzerland! It's been months since I met a new country.)
[F 2] [R 0] @MariamGhazalaa      I just wanna go back to Syria
-----
Sun Oct 20 17:09:51 +0000 2013 @ESP_09 favored:
[F 1] [R 1] @tedfeed             ted video HD TED: Retain Patel: Who am I? Think again - Retain Patel / Yuyu Rau (2013): How do we decide who w... http://t.co/u1DMOX8KnI
[F 156] [R 710] @KinDotcom          Fact: Gmail, iCloud, Skype, etc. have to provide (by law) secret & untraceable NSA backdoors to all your data. #GetOutNow
[F 64] [R 35] @r_netsec           Penetration Testing Practice Labs .. LOTS of them! - http://t.co/mNEaXGRCAy
[F 36] [R 12] @Kailichi           @TheBloggess do you own these shoes? If not you need them . They fit your chicken fetish. http://t.co/oBzR4FAlDs
[F 3] [R 5] @MrsYisWhy           My presentation from @BSidesDC: Homunculus: Why You Will Lose the Battle of BYOD&gt; @slideshare http://t.co/g3tQu9GbDF
[F 242] [R 68] @lifehacker           Looking to learn a new language? We have five great ways to pick one up: http://t.co/lBuTqvgJGw
[F 11] [R 3] @r_netsec           TinyShell - Ported to SCTP (code + video inside) - http://t.co/PwjD4X8CGf
[F 18] [R 34] @donmcmillan          BIG DATA Truths: Coffee price at Starbucks is directly related to Engineer's starting salary @EngineersHumor http://t.co/LlkOsyPdSc
[F 1] [R 3] @irongeeek_adc         NSA Wiretaps are Legal and Other Annoying Facts - @cowboysfaninky Branden Miller for @Hack3rcon 4 http://t.co/4cUJhlWy6M
[F 25] [R 31] @SecurityTube        Decrypt SSL Traffic using private key files : http://t.co/k3VFhSYD9w
[F 19] [R 10] @DavidJBianco         Slides for my "Enterprise Security Monitoring" talk at #BSidesDC this morning can be found at http://t.co/9AStWOQvhG. Feedback encouraged!
[F 1] [R 1] @CompuSecure           Dick Cheney Altered Heart Implant to Thwart Wireless Hackers http://t.co/E6holOgp9U
[F 23] [R 4] @r_netsec           Watch How NOT to do a Penetration Test | SOURCE Barcelona 2011 Episodes | Videos - http://t.co/MR4ipvRE9r
[F 11] [R 9] @r_netsec           Facebook CSRF leading to full account takeover - http://t.co/IMpHjyCo5v
[F 26] [R 62] @hdmooore           Source code for the backdoored GoAhead web server in Tenda is on GitHub: https://t.co/0yFq3JwjQ8 (found by Preston)
-----
Sun Oct 20 14:18:38 +0000 2013 @taoeffect favored:
[F 146] [R 517] @_wirepair      Congratulations on your device purchase!
Would you like:
A. A US Backdoor.
B. A Chinese Backdoor.
C. A Vendor Backdoor.
-----
```


STTLibary @sttlibrary · 2m
stttwitter_cocoaheads.003.jpg



View more photos and videos

STTLibary @sttlibrary · 2m
stttwitter_cocoaheads.002.jpg



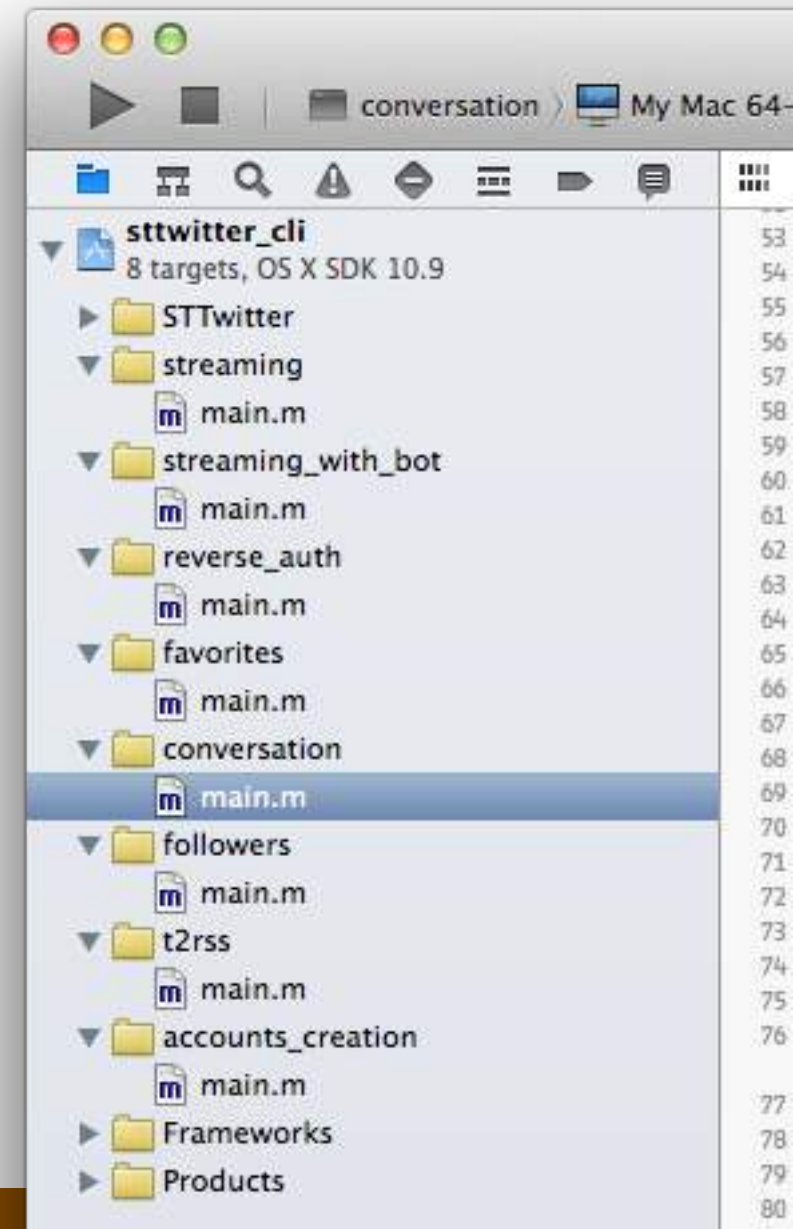
View more photos and videos

STTLibary @sttlibrary · 2m
stttwitter_cocoaheads.001.jpg



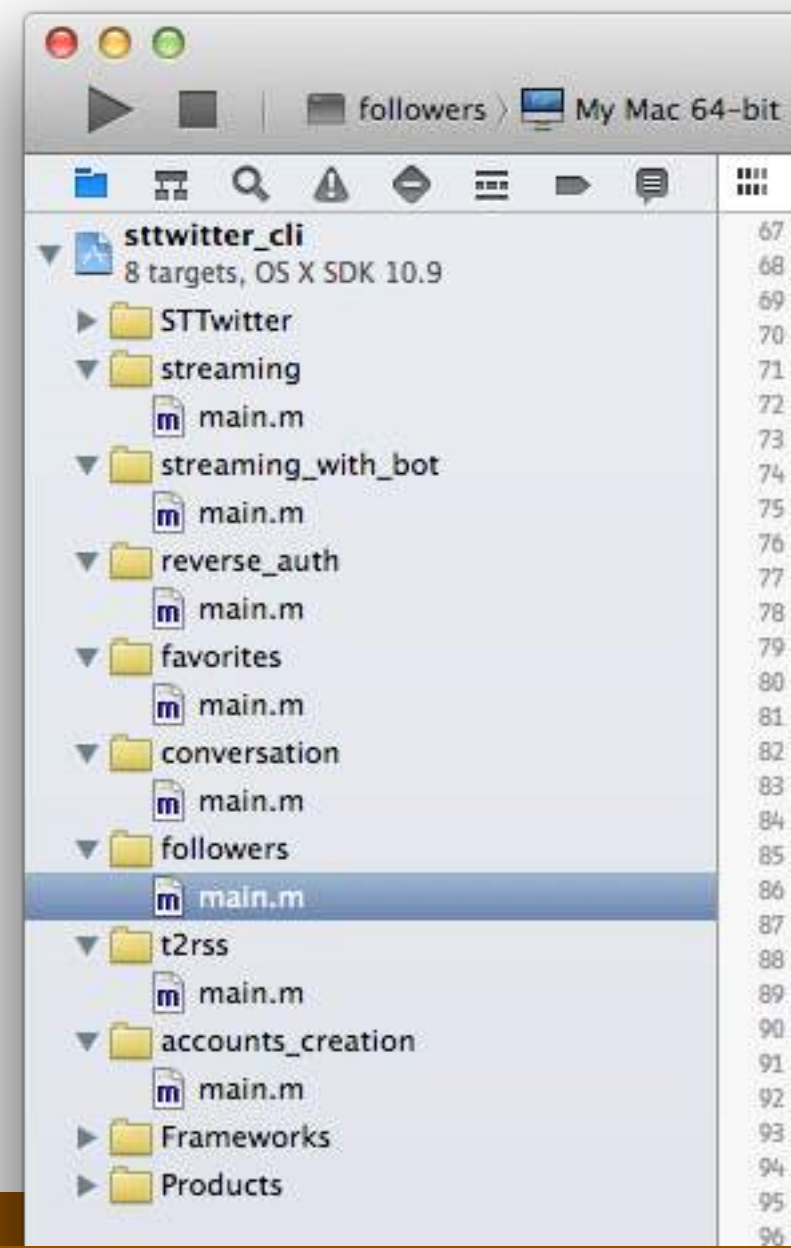
View more photos and videos

posting images
in the same
conversation



```
All @SbClx Followers
-- users: 20
-- previousCursor: 0
-- nextCursor: 1359665742523042866
-- wait for 0 seconds
-- users: 2
-- previousCursor: -1336693308248278824
-- nextCursor: 0
-- allFollowers: (
    snr021,
    java4Noobs,
    ClaireReynaud,
    SebastienHugues,
    geraudch,
    "pasc_r",
    FlowSwiss,
    EskandarMounif,
    Lergonom,
    CorinaCroitoru,
    cjwl,
    hdep,
    "pjay_",
    MochaCode,
    CocoaLausanne,
    applyticsapp,
    dearprakash,
    NSZurich,
    bontoJR,
    0xcd,
    un23soleil,
    nst021
)
```

using cursors





- Today's Articles
- Unread Articles
- Marked Articles
- Vienna Support 39
- Apple Hot News 20
- Vienna Developer Blog 10
- Mac OS X Hints 20
- BBC World News 104
- World of Apple 60
- The Unofficial Apple Weblog (TUAW) 50
- Trash
- nst021 88**

Filter by: All Articles

Subject	Date	Author
RT @IrisKV: C'est assez inquiétant,mais je crois que le perso de c...	Today at 10:50	@pyknite
Thanks to @N1aKan for the awesome logo on my new website ;)...	Today at 10:44	@r00tbsd
If you have interesting #malware for me, you can use my new sit...	Today at 10:43	@r00tbsd
Reminder: I'll be presenting the STTwitter library tonight at CocoaHe...	Today at 10:40	@nst021
RT @cBekrar: Kudos to @thezdi for sending a big "fuck you" to v...	Today at 10:31	@Myst3rie
RT @ReidParker_: .@codinghorror @Brocklesnitch Wolfenstein ga...	Today at 10:22	@angealbertini
RT @codinghorror: Wolfenstein game graphics, 1992 vs 2014 http://...	Today at 10:21	@newsoft
Treasure Map(Kit) by @arepty from NSConf 6 is now online at htt...	Today at 10:13	@nsconf
RT @HockeyInJune: CFP deadline for @USENIXSecurity WOOT '14...	Today at 10:08	@Myst3rie
I use mocks and I'm happy with that http://t.co/oA58NYftJc	Today at 09:56	@secboffin
RT @cedricpernet: RT @ncaproni: eConflicts: Des militaires chinois re...	Today at 09:56	@sud0man
No, I will not vote for any of those entitled morons today. I alrea...	Today at 09:52	@secboffin
RT @HakanEkizer: Metasploit: Controlling Internet Explorer user...	Today at 09:46	@sud0man
RT @i0n1c: @angealbertini no now we have 45k of scared people...	Today at 09:22	@angealbertini
RT @fabsx00: Our S&P paper/slides on Modeling and Discoverin...	Today at 09:11	@JonathanSalwan
my AngeCryption slides got 45K views - I hope I made crypto &...	Today at 09:10	@angealbertini

Reminder: I'll be presenting the STTwitter library tonight at CocoaHeads Lausanne <http://t.co/LUT2Bmp6EO> 20:15 at Restaurant Le Milan

From: nst021

Date: Today at 10:40

Enclosure:

Reminder: I'll be presenting the STTwitter library tonight at CocoaHeads Lausanne <http://t.co/LUT2Bmp6EO> 20:15 at Restaurant Le Milan

@nst021

iOS 7 Accounts Creation

Consumer key: IHUYavQ7mmPBhNiBB1F9Q

Consumer secret: cIBZ*****

Access token: 179654598-yukdZLZcDfxU5PZBZdJcCpaJF5bKUJTdxXoxUZ9u

Access secret: YEhw*****

POST <https://api.twitter.com/1/account/generate.json>

adc:	pad
discoverable_by_email:	0
email:	EMAIL
geo_enabled:	0
lang:	en
name:	NAME
password:	PASSWORD
screen_name:	SCREEN_NAME
send_error_codes:	true
time_zone:	CEST

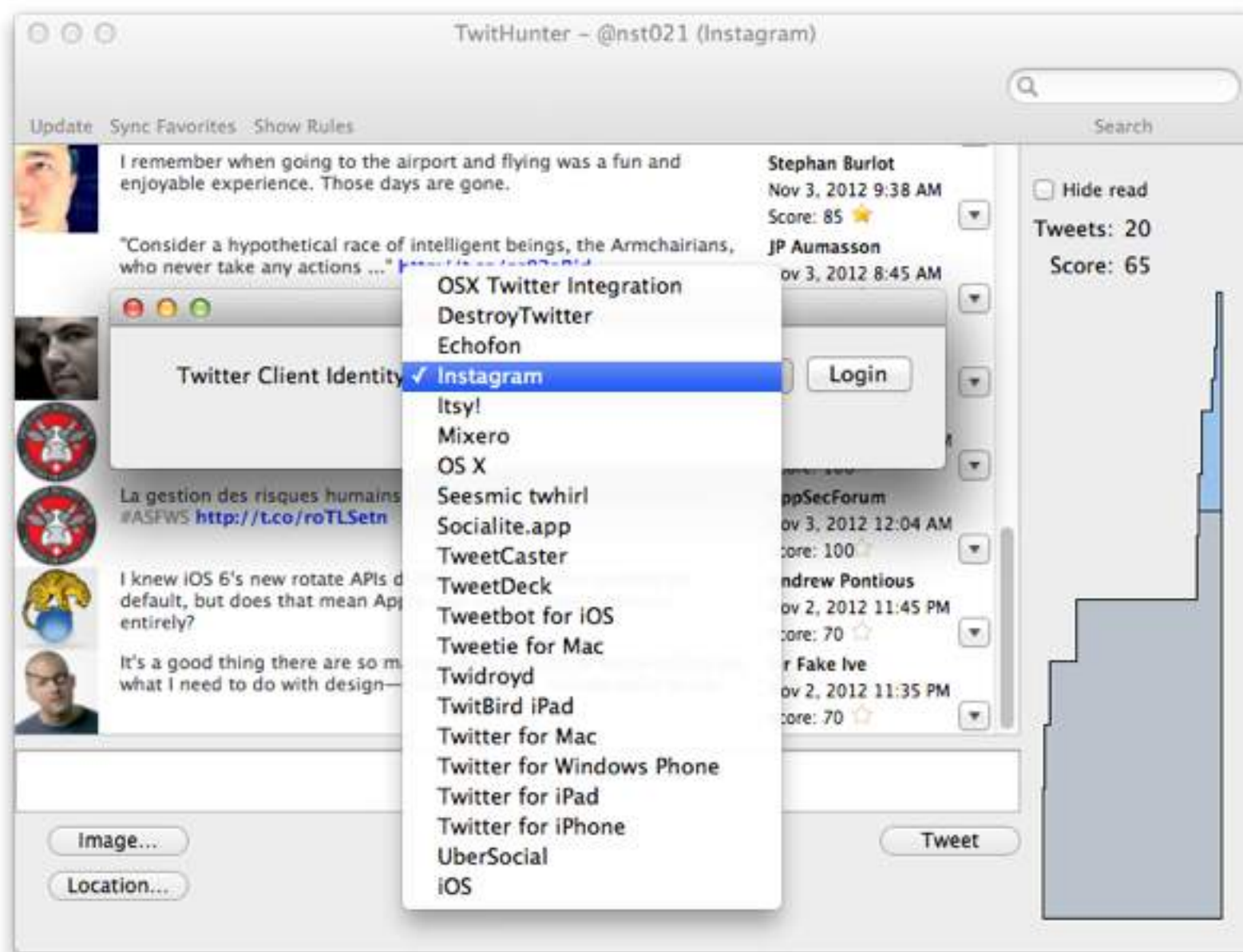


demo

Debug

```
@Myst3rie      + RT @msuiche: We are in a new Era where journalists claim to
May 22 12:10:46 + be hackers, and where technical experts are CEH certified.
+
@newsoft      L'#ANSSI fait la promotion de ADA et OCaml
May 22 12:03:24 http://t.co/VKHuzUhd8b mais ne parle pas de Go
http://t.co/Fec9R0mVfp tsss... ;)
@lolopb      RT @counternotions: "?~@?which fits in part with Google's
May 22 11:47:12 mission to provide internet access to the entire world."
(That's not Google's "mis?~@?
@zisyadis    Thessalonique: seule ville européenne de plus d'un million
May 22 11:36:56 d'habitants... Sans macdo.
@cybersmashup RT @HackRead: #eBay and #Paypal Hacked; 128 Million Users
May 22 11:24:44 Asked to Change Passwords. MORE ?~^~\ http://t.co/24Bgs72Ml4
#Security http://t.co/rl?~@?
@cybersmashup It takes some time... Microsoft hasn?~@~Yt disclosed neither
May 22 11:22:41 fixed a 4 months-old #0Day Internet Explorer 8 flaw
http://t.co/p6pvSy8XFJ
```


TwitHunter



<https://github.com/nst/TwitHunter>

Useful Categories

```
// use default values for URL shortening
-[NSString st_numberOfCharactersInATweet];

// "created_at" = "Tue Jul 31 07:58:03 +0000 2012"
+[NSDateFormatter st_TwitterDateFormatter];

// http://t.co/URrfrKZptG
+[STHTTPRequest expandedURLStringForShortenedURLString:
                                     successBlock:
                                     errorCallback:];
```

Adding Categories

STTwitter

```
/*  
GET    lists/subscribers/show  
  
Check if the specified user is a subscriber of the specified list.  
Returns the user if they are subscriber.  
*/  
  
- (void)getListsSubscribersShowForSlug:(NSString *)slug  
    ownerScreenName:(NSString *)ownerScreenName  
    orOwnerID:(NSString *)ownerID  
    userID:(NSString *)userID  
    orScreenName:(NSString *)screenName  
    includeEntities:(NSNumber *)includeEntities // @(YES)  
    skipStatus:(NSNumber *)skipStatus // @(NO)  
    successBlock:(void (^)(void))successBlock  
    errorCallback:(void (^)(NSError *error))errorBlock;
```

STTwitter+YourApp

```
- (void)getListsSubscribersForListName:(NSString *)listName  
    successBlock:(void (^)(void))successBlock  
    errorCallback:(void (^)(NSError *error))errorBlock;
```

Deeper, Generic Fetch Method

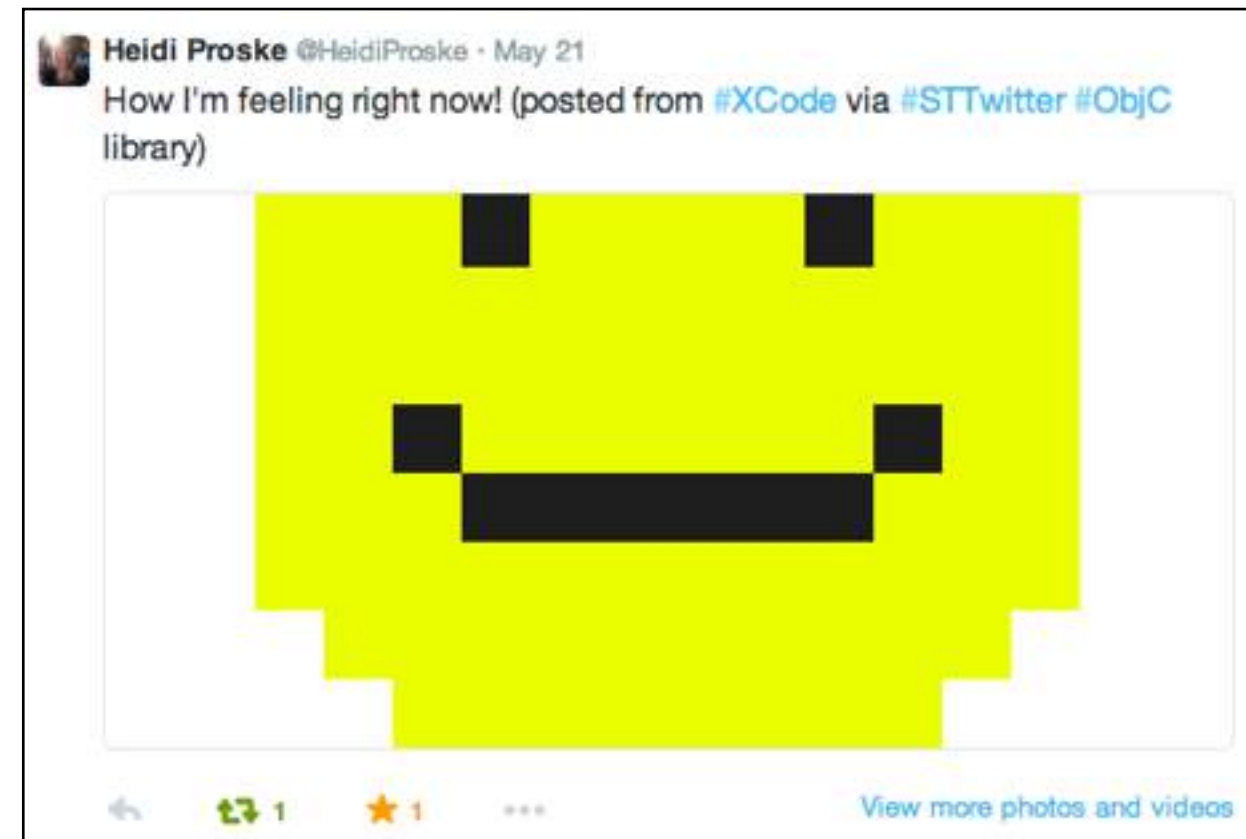
```
- (id) fetchResource:(NSString *)resource
    HTTPMethod:(NSString *)HTTPMethod
    baseURLString:(NSString *)baseURLString
    parameters:(NSDictionary *)params
uploadProgressBlock:(void (^)(NSInteger bytesWritten,
                               NSInteger totalBytesWritten,
                               NSInteger totalBytesExpectedToWrite))uploadProgressBlock
downloadProgressBlock:(void (^)(id request,
                               id response))downloadProgressBlock
successBlock:(void (^)(id request,
                      NSDictionary *requestHeaders,
                      NSDictionary *responseHeaders,
                      id response))successBlock
errorBlock:(void (^)(id request,
                    NSDictionary *requestHeaders,
                    NSDictionary *responseHeaders,
                    NSError *error))errorBlock;
```

demo followers / rate limits

Contributions

- Some dude with ` & ` in his password :-)
- GitHub issues
- StackOverflow questions with `sttwitter` tag
- Not always easy to deal with pull requests

Enthusiastic Users



<https://twitter.com/nilsou/status/392364862472736768>

<https://twitter.com/HeidiProske/status/468909661849399296>

Also integrated in Adium, Osfoora, ...



Recap

- Security issues: HITBAMS2013, ASFWS12/13/14
- SLComposeViewController and SLRequest
- STTwitter unlocks more flows:
XAuth, Reverse auth, Web auth, App only, ...
- STTwitter maps endpoints / Obj-C methods
- Stories on writing an Objective-C library