

Twitter Integration

Tweet, tweet

Session 124

Glen Steele and Scott Herz

iOS Software Engineers

These are confidential sessions—please refrain from streaming, blogging, or taking pictures



I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

I will not instigate revolution.

Bart Simpson

Twitter Stats



- 200 million users
- One billion tweets per week
- 5,530 TPS during Japan earthquake

3,000

iOS Apps with Twitter integration

What You Will Learn

- Pain-free Tweeting
- How to access built-in iOS Twitter accounts
- How to interact with the Twitter API

Tweet Sheet

Pain-free Twitter integration



Presentation

TWTweetComposeViewController

- Check for service availability
- Create a view controller
- Supply a completion handler
- Present modally



Presentation

TWTweetComposeViewController

```
//MyViewController.m
```

```
- (void)loadView {  
    if ([TWTweetComposeViewController canSendTweet]) {  
        //show my tweet button  
        _showTweetButton = YES;  
    }  
}
```

```
    //load my views...  
}
```

```
- (void)tweetButtonTapped:(id)sender {  
}
```

Presentation

TWTweetComposeViewController

```
- (void)tweetButtonTapped:(id)sender {  
}  
    TWTweetComposeViewController *tweetSheet =  
        [[TWTweetComposeViewController alloc] init];  
  
    tweetSheet.completionHandler =  
    ^(TWTweetComposeViewControllerResult result) {  
        [self dismissModalViewController:tweetSheet animated:YES];  
    };  
  
    [self presentModalViewController:tweetSheet animated:YES];  
    [tweetSheet release];  
}
```

Attachments



- Text
 - `(BOOL)setInitialText:(NSString *)text;`
- Images
 - `(BOOL)addImage:(UIImage *)image;`
- URLs
 - `(BOOL)addURL:(NSURL *)url;`

Sizing

- 140 character maximum
- Images and URLs use characters
 - Currently 19 characters
- URL lengths could change; use return BOOLS!

Demo

Scott Herz

Senior iOS Software Engineer

Tips and Tricks

- Only designed for modal presentation

```
[self presentViewController:tweetSheet animated:YES];
```

- Set properties prior to presentation
- Check `+(BOOL)canSendTweet`
- Large images affect performance

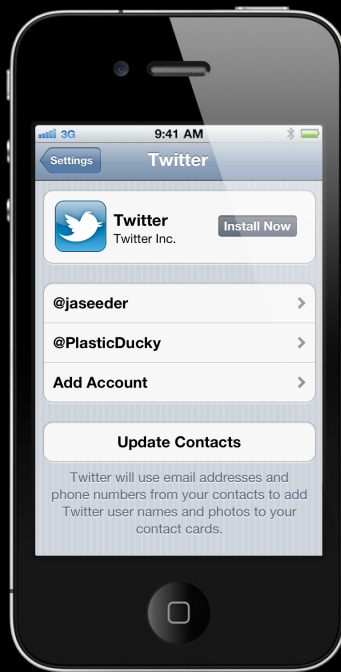
Tweet Sheet

Pain-free Twitter integration



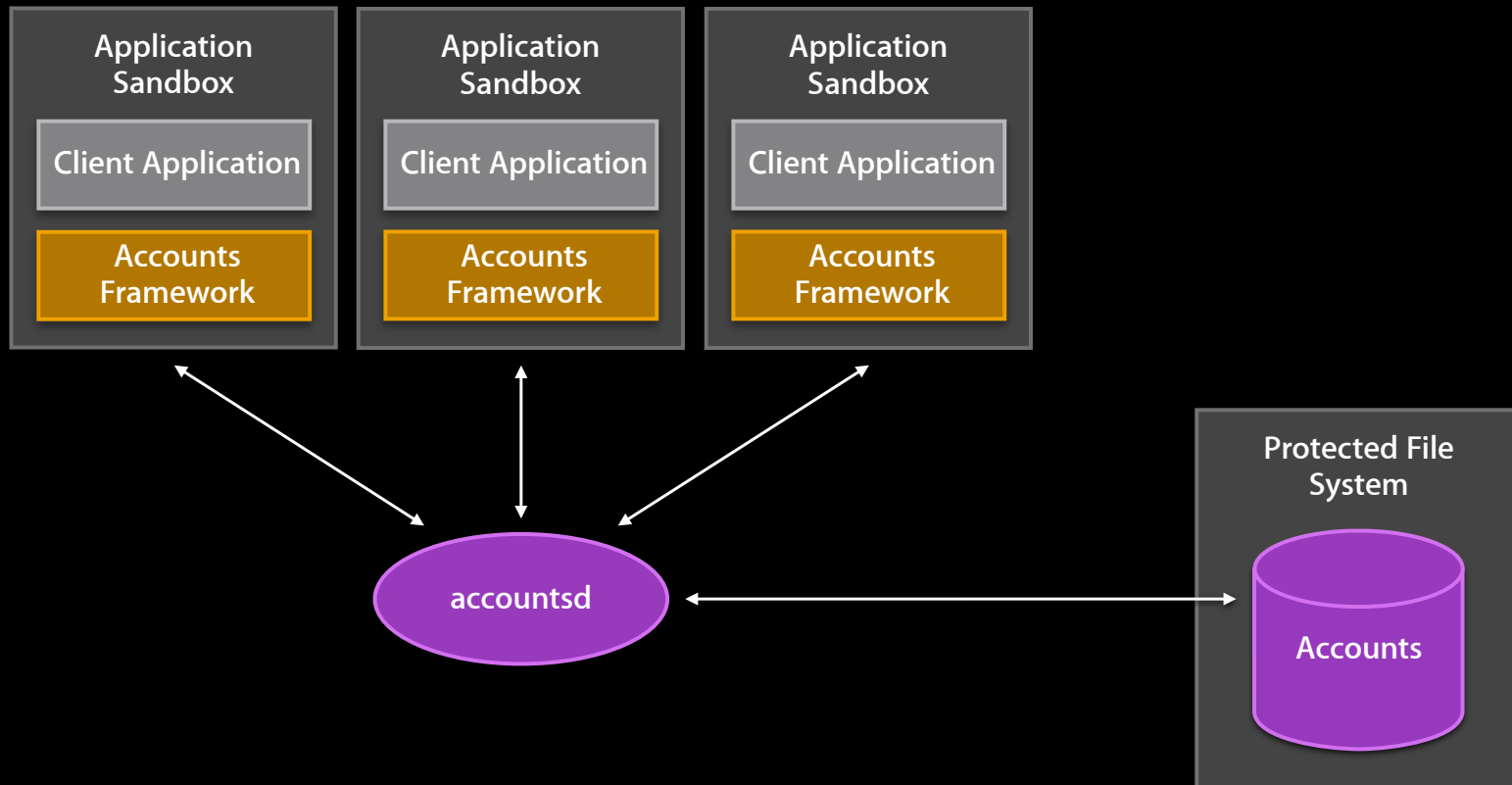
Accounts

Accounts Framework

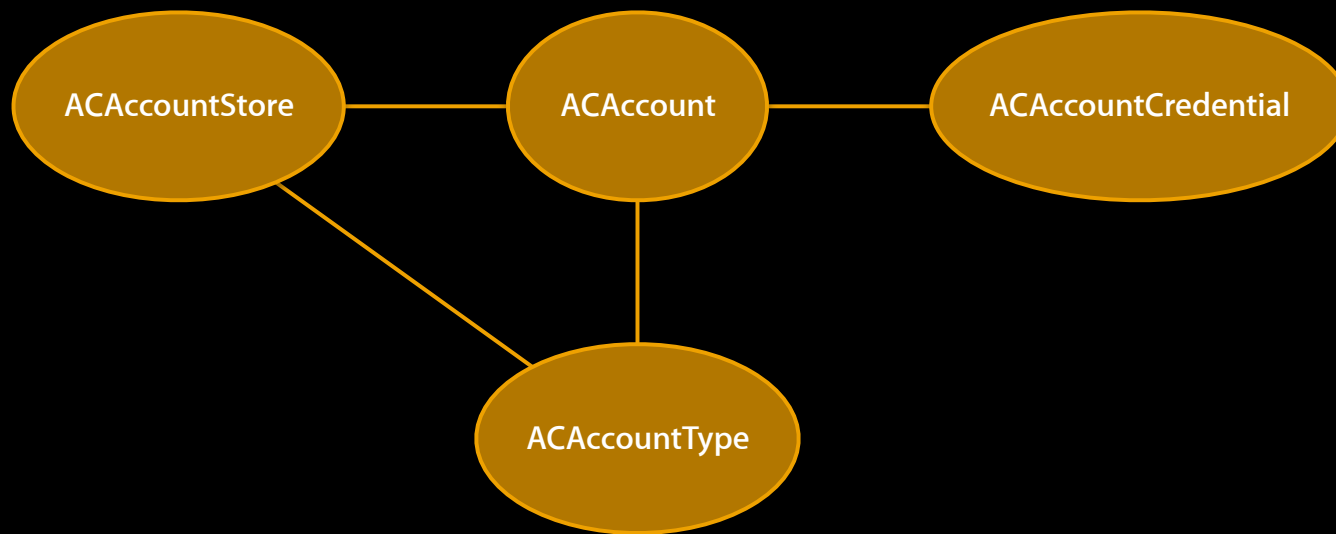


- Access to built-in system accounts
- Twitter only
- Add accounts
- Automatic validation

Architecture



Classes



ACAccountStore

- Your connection to the database
- Generally long-lived instances
- Accounts are bound to the store they were fetched from
- ACAccountsChangedNotification

ACAccount

- Represents the system account
- Properties
 - Identifier
 - Account type
 - Username
 - Credential
 - Description

ACAccountType

- Every account has an account type
- Properties
 - Type identifier
 - Description
 - Access granted

Accessing Accounts

Request permission

- Accounts are not accessible by default
- Protected by account type

```
ACAccountStore *store = [[ACAccountStore alloc] init];

ACAccountType *twitterType = [store
    accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierTwitter];

[store requestAccessToAccountsWithType:twitterType
    withCompletionHandler:^(BOOL granted, NSError *error) {
    if (granted) {
        //access granted
    }
}];

[store release];
```

Accessing Accounts

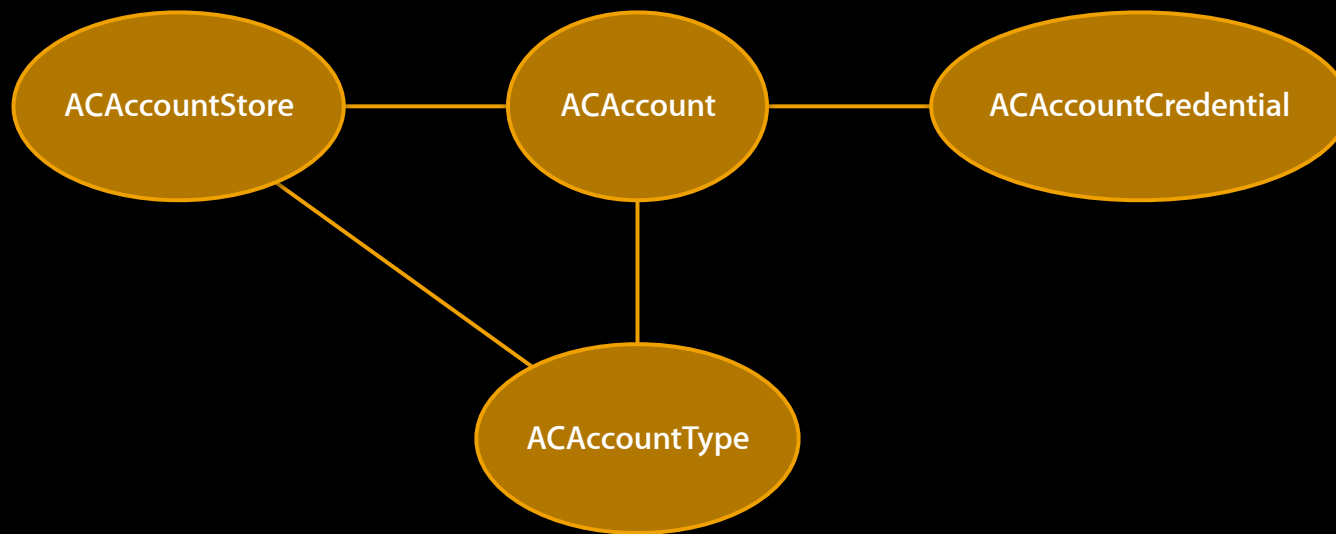
- Request by account type

```
ACAccountType *twitterType = [store  
    accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierTwitter];  
NSArray *twitterAccounts = [store accountsWithAccountType:twitterType];
```

- Request by account identifier

```
NSString *savedIdentifier = account.identifier;  
//later on...  
ACAccount *account = [store accountWithIdentifier:savedIdentifier];
```


Classes



Creating New Accounts

- Add new system Twitter accounts from your app
 - Migrate OAuth credentials
- Twitter accounts support automatic validation

Creating New Accounts

Migrating OAuth credentials

```
ACAccount *account = [[ACAccount alloc] initWithAccountType:twitterType];
ACAccountCredential *oauthCredential =
    [[ACAccountCredential alloc] initWithOAuthToken:token
                                     tokenSecret:secret];

account.credential = oauthCredential;
[accountStore saveAccount:account withCompletionHandler:^(BOOL success,
NSError *error) {

    if (success) {
        //account validated and saved
    } else {
        //save failed, handle your errors!
    }

}];

[oauthCredential release];
[account release];
```

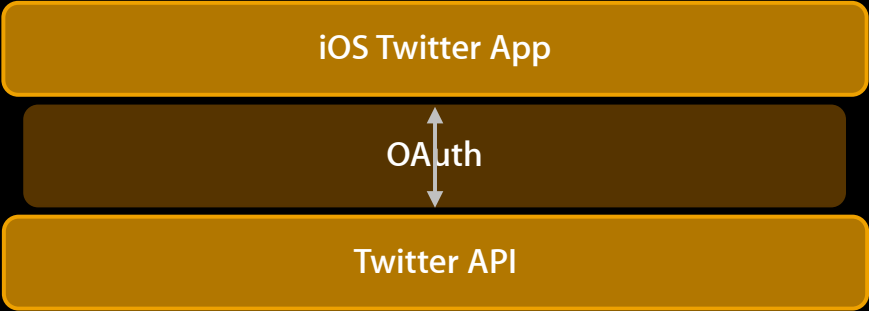
Tips and Tricks

- Ask for access when you need it
- Certain operations always gated with user alerts
- Do not cross the streams!
- Pay attention to `ACAccountsChangedNotification`
- Completion handlers called on arbitrary thread
 - UI: `dispatch_async(dispatch_get_main_queue(), ^{//do stuff});`

Accounts



Twitter Request API



TWRequest



- Not an Objective-C abstraction of Twitter's API
- Quick and easy arbitrary Twitter requests
- We take care of the OAuth signing
- Use any Twitter account
 - Must be granted access
- Support for multipart data

TWRequest

Basic properties

URL

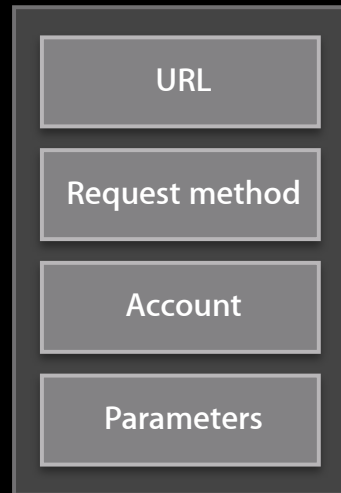
Request method

Account

Parameters

TWRequest

Basic properties



GET statuses/home_timeline | dev.twitter.com

URL

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

Supported formats

json, xml, rss, atom

Supported request methods

GET

Requires Authentication

true [About authentication »](#)

Rate Limited

true [About rate limiting »](#)

Parameters

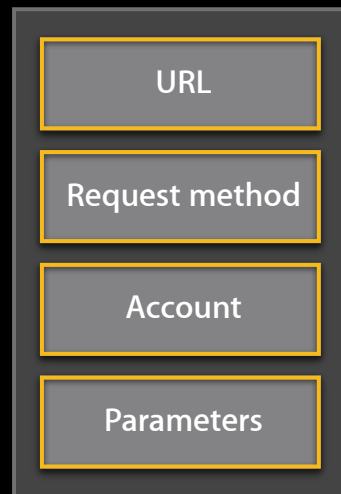
Optional

- **since_id**
Returns results with an ID greater than (that is, more recent than) the specified ID. There the number of Tweets which can be accessed through the API. If the limit of Tweets has c since the since_id, the since_id will be forced to the oldest ID available.
`http://api.twitter.com/1/statuses/home_timeline.json?since_id=12345`
- **max_id**
Returns results with an ID less than (that is, older than) or equal to the specified ID.
`http://api.twitter.com/1/statuses/home_timeline.json?max_id=54321`

March 30, 2011 Bring interactivity to Tweets that you display on the web with Web Intents. x

TWRequest

Basic properties



GET statuses/home_timeline | dev.twitter.com

URL

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

Supported formats

json, xml, rss, atom

Supported request methods

GET

Requires Authentication

true [About authentication >](#)

Rate Limited

true [About rate limiting >](#)

Parameters

Optional

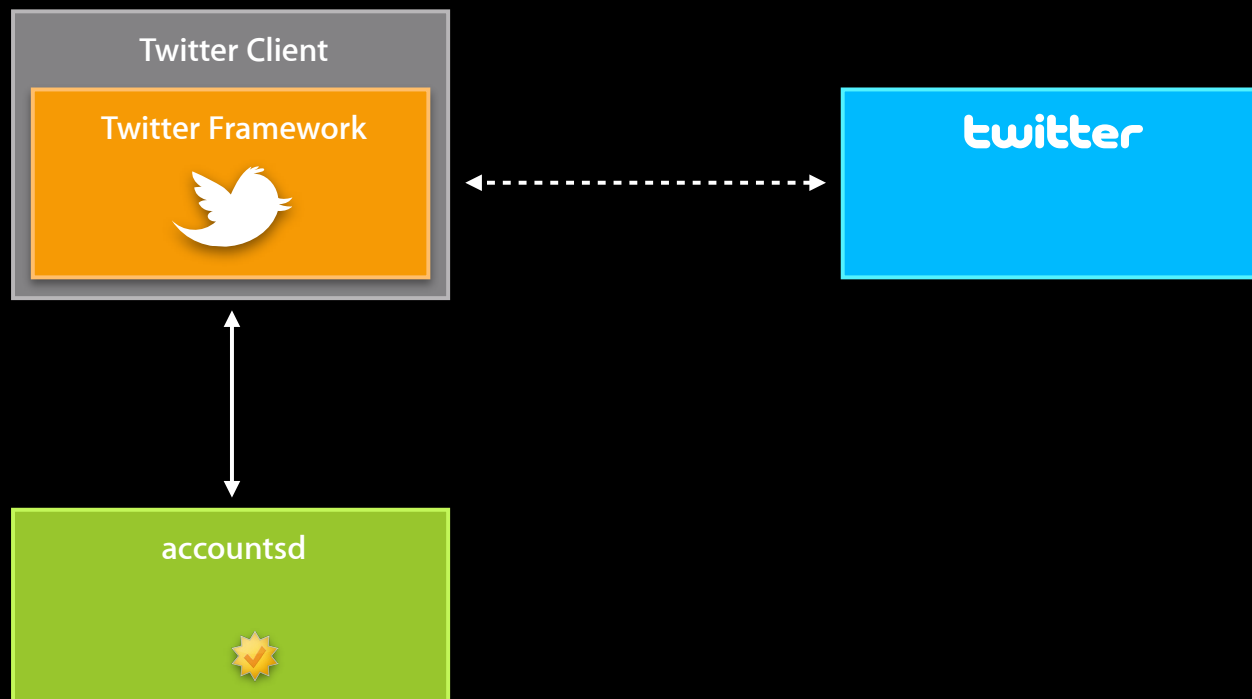
- since_id**
Returns results with an ID greater than (that is, more recent than) the specified ID. There the number of Tweets which can be accessed through the API. If the limit of Tweets has c since the since_id, the since_id will be forced to the oldest ID available.
`http://api.twitter.com/1/statuses/home_timeline.json?since_id=12345`
- max_id**
Returns results with an ID less than (that is, older than) or equal to the specified ID.
`http://api.twitter.com/1/statuses/home_timeline.json?max_id=54321`

March 30, 2011 Bring interactivity to Tweets that you display on the web with Web Intents. x

Performing Requests

- Call `performRequestWithHandler:`
 - Single atomic request
 - Result data is returned
- Request a signed NSURLRequest: `-(NSURLRequest *)signedURLRequest;`

OAuth Signing



Using TWRequest

```
NSURL *timeLineURL= [NSURL URLWithString:@"http://api.twitter.com/1/statuses/
home_timeline.json"];

NSDictionary *parameters = [NSDictionary dictionaryWithObjectsAndKeys:
                            @"10", @"count", nil];

TWRequest *myTimeLine = [[TWRequest alloc] initWithURL:timeLineURL
                            parameters:parameters
                            requestMethod:TWRequestMethodGET];

[myTimeLine performRequestWithHandler:^(NSData *responseData,
NSHTTPURLResponse *urlResponse,
NSError *error) {

//handle stuff

}];
```

Using TWRequest

```
[myTimeLine performRequestWithHandler:^(NSData *responseData,
                                         NSHTTPURLResponse *urlResponse,
                                         NSError *error) {

    //handle stuff
    NSError *jsonError = nil;
    id timelineData = [NSJSONSerialization JSONObjectWithData:responseData
                                                         options:0
                                                         error:&jsonError];

    //deal with your errors!

    dispatch_async(dispatch_get_main_queue(), ^{
        [self updateTimeline:timelineData];
    });

}];
```

Multipart Data

```
NSURL *profileURL = [NSURL URLWithString:@"http://api.twitter.com/1/account/
update_profile_image.json"];

TWRequest *updateProfile = [[TWRequest alloc] initWithURL:profileURL
                                                    parameters:nil
                                                    requestMethod:TWRequestMethodPOST];

[updateProfile addMultiPartData:UIImageJPEGRepresentation(pickedImage, 0.7)
                withName:@"media"
                type:@"image/jpeg"];

[updateProfile performRequestWithHandler:^(NSData *responseData,
                                           NSHTTPURLResponse *urlResponse,
                                           NSError *error) {

    //handle stuff

}];
```


Demo

Monkeys



- Shrinking habitat
- Often addicted to cigarettes
- No access to Twitter

Demo

Tips and Tricks

- Follow the Twitter “Rules of the Road”
- Completion handlers called on arbitrary thread
 - UI: `dispatch_async(dispatch_get_main_queue(), ^{//do stuff});`

Twitter



More Information

Bill Dudney

Application Framework Evangelist
dudney@apple.com

Documentation

iOS Twitter and Accounts Frameworks
<http://developer.apple.com/>

Twitter API

<http://dev.twitter.com/doc>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Blocks and Grand Central Dispatch in Practice

Pacific Heights
Wednesday 10:15–11:15AM

Core OS Networking, Key Principles

Marina
Tuesday 9:00–10:00AM

Labs

Twitter and Accounts Lab

Application Frameworks Lab D
Thursday 4:30–6:00PM

Summary

- Tweet Sheet
- Accounts
- TWRequest



