Twitter Application Authentication With iOS: Or How I Stopped Worrying and Learned to Love URI-Encoded Meaningless Strings

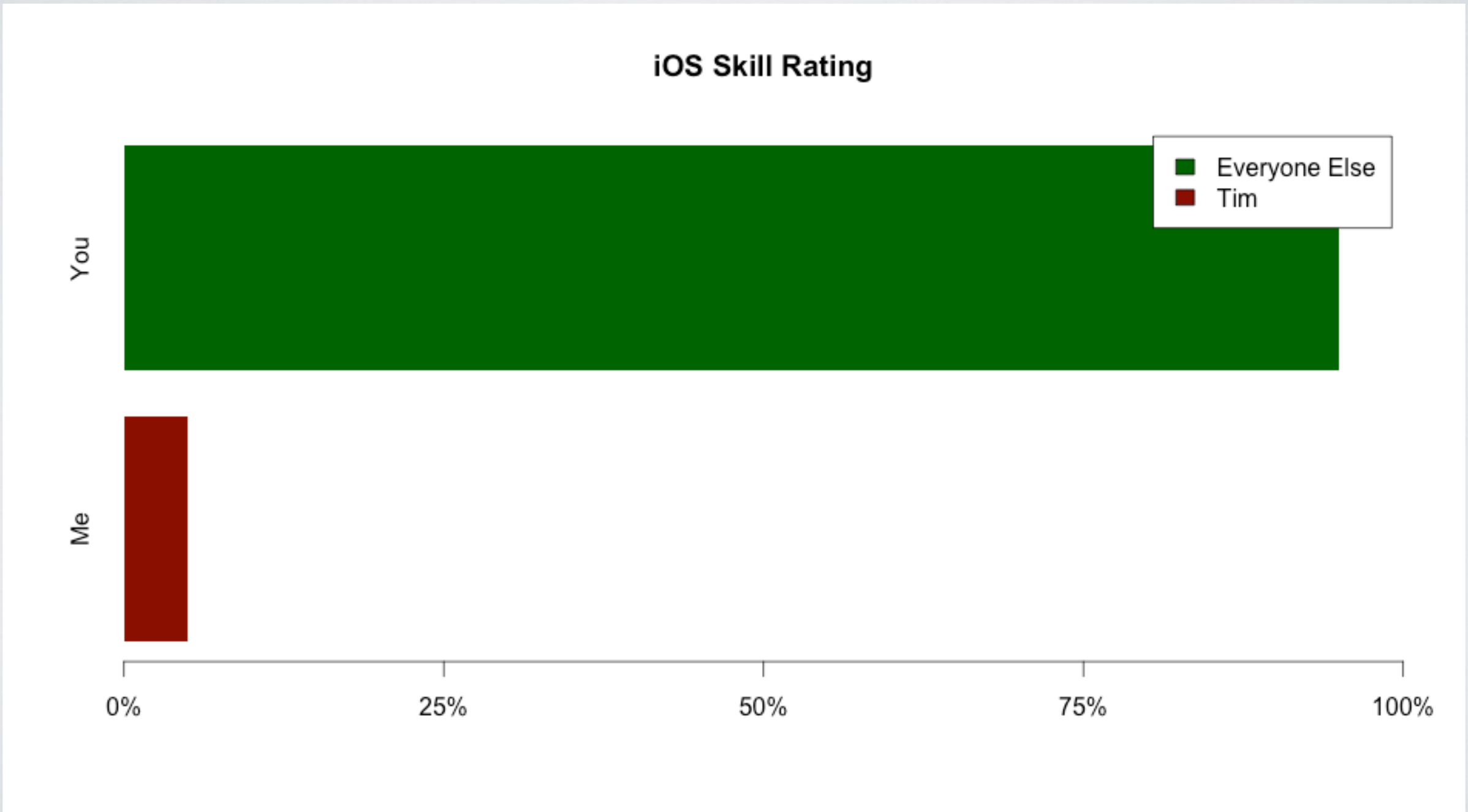
Tim Hoolihan
@thoolihan

Who Am I

- Use: Ruby, .Net, JS(Angular, Backbone), R, Phonegap, iOS and more

- Work: http://lvlsvn.com @lvlsvn We're hiring

Disclaimer*

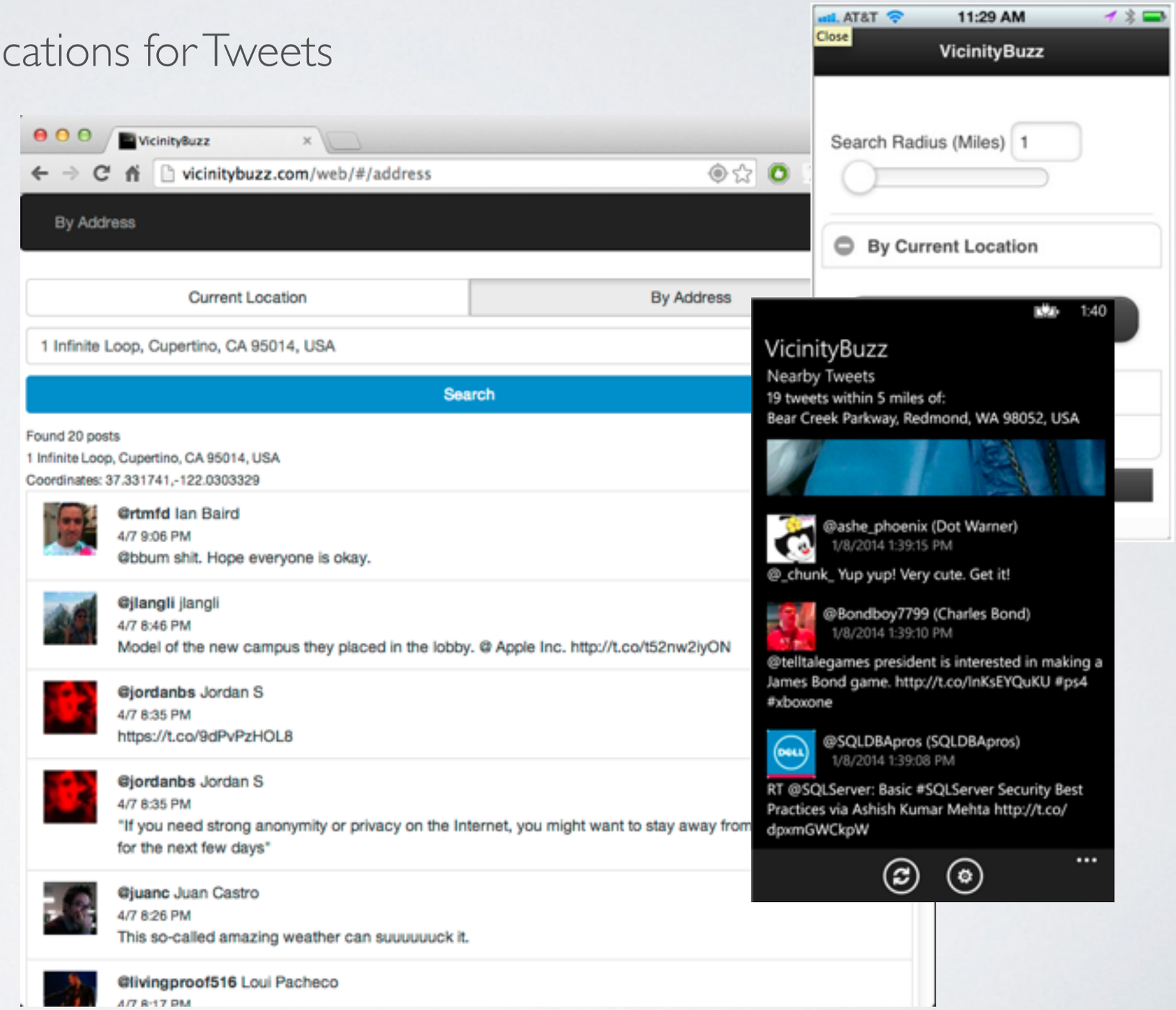


Slides Available

- http://hoolihan.net/blog-tim/slides

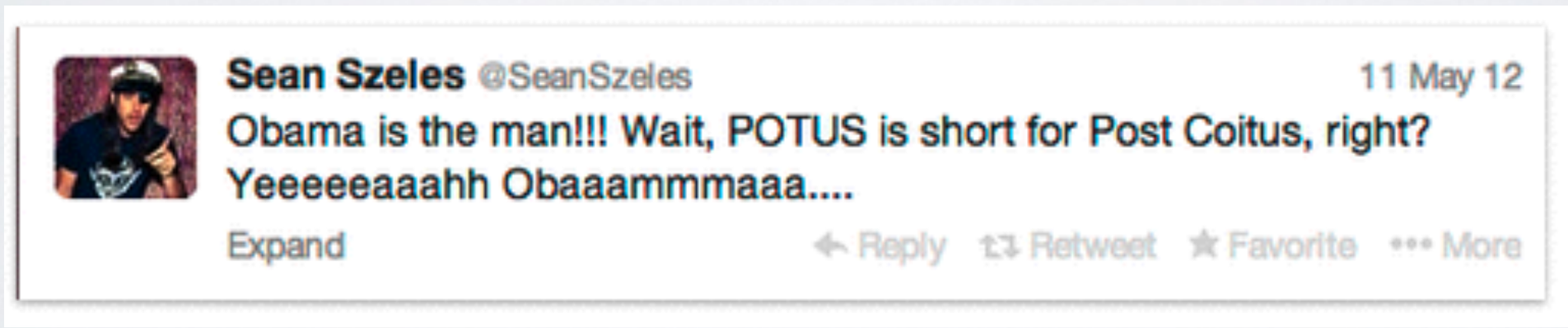
What is VicinityBuzz

- Search Geographic Locations for Tweets
  - Nearby
  - By Address
- Why?



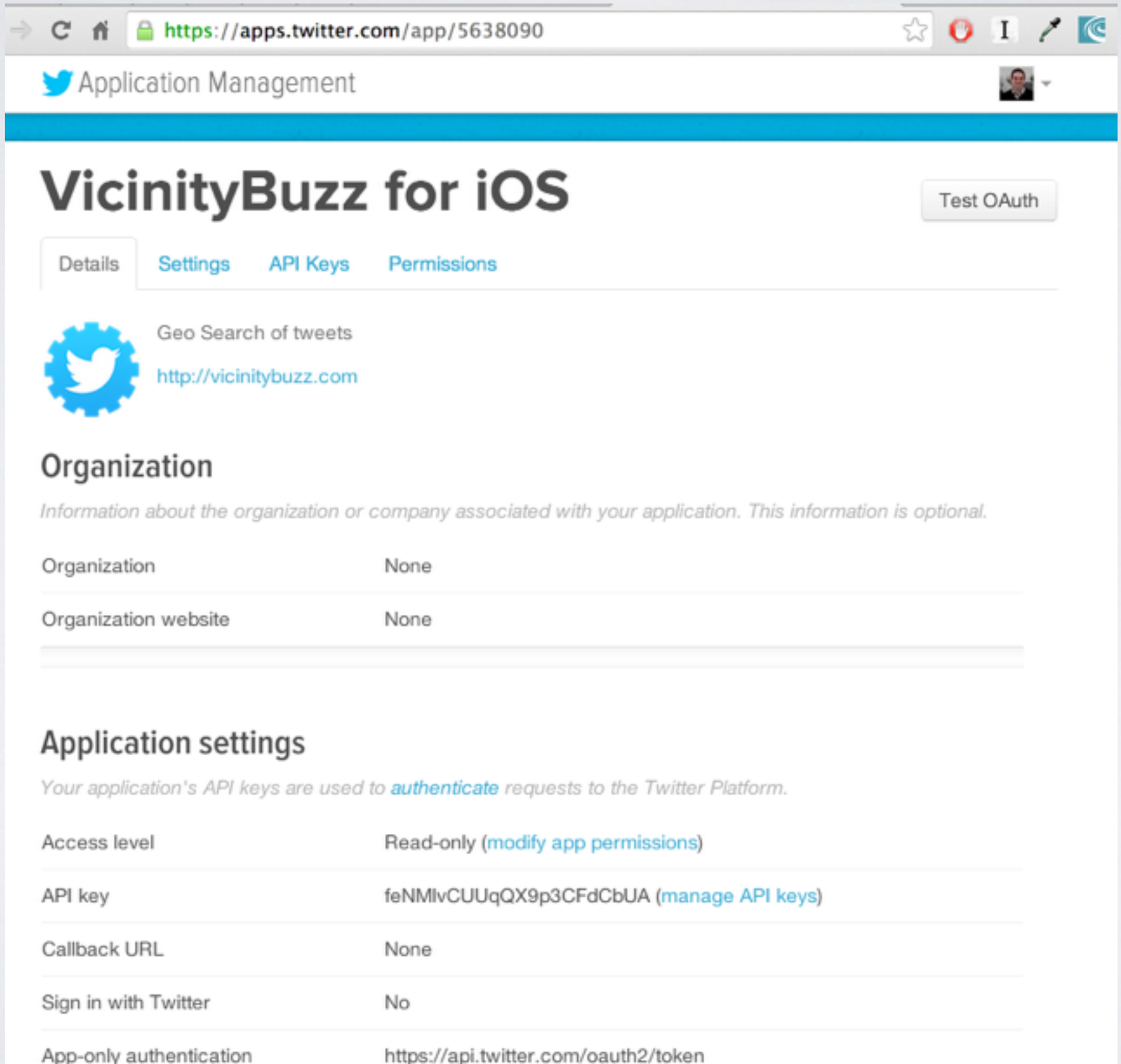
Demo

- Warning: There is offensive and un-intelligent content on twitter… who knew?
- So you could see something like…

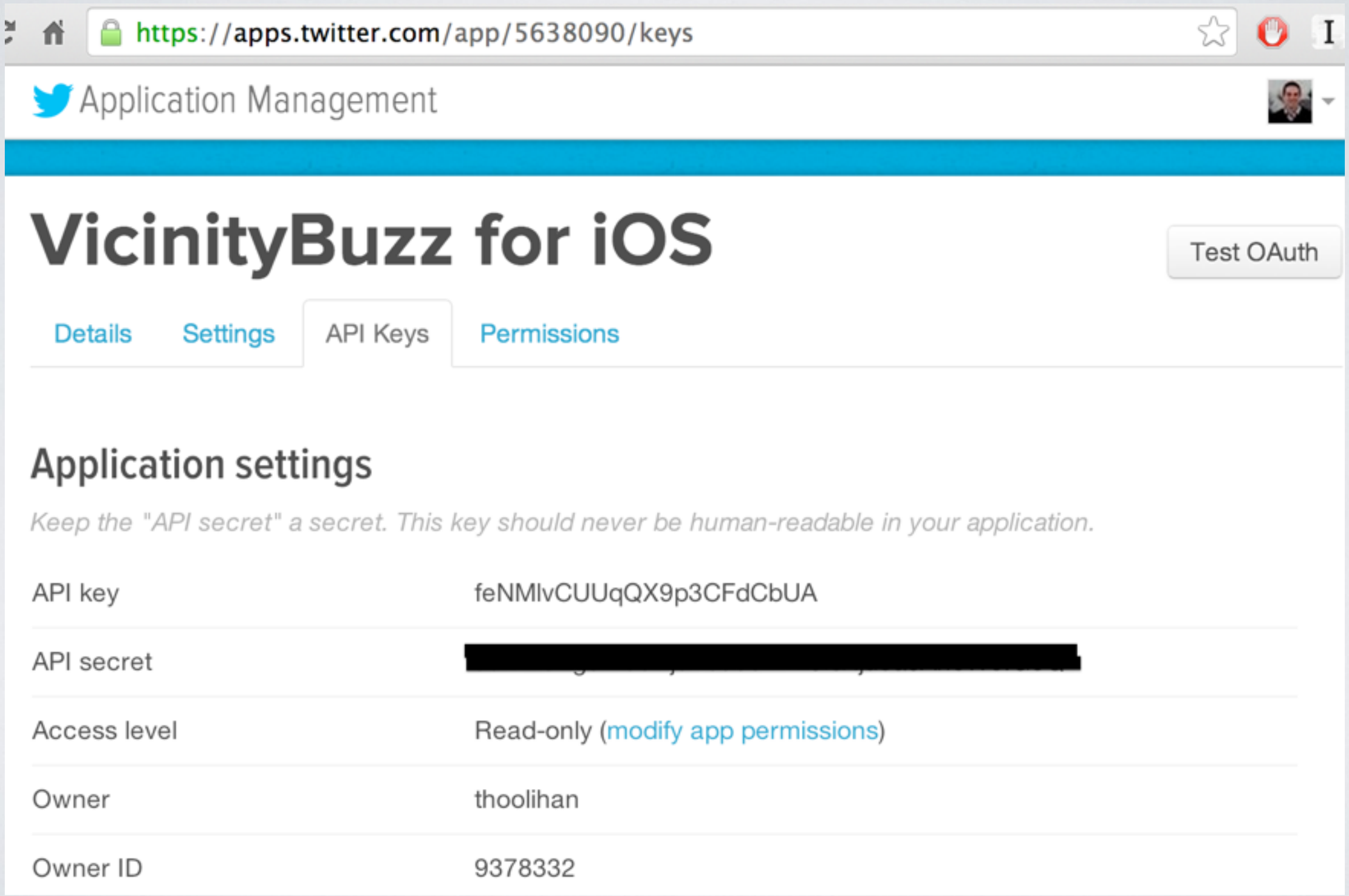


http://dumbesttweets.com

dev.twitter.com



Obtaining API Keys



Preparing Params

Consumer key	xvz1evFS4wEEPTGEFPHBog
Consumer secret	L8qq9PZyRg6ieKGEKhZolGC0vJWLw8iEJ88DRdyOg
RFC 1738 encoded consumer key (does not change)	xvz1evFS4wEEPTGEFPHBog
RFC 1738 encoded consumer secret (does not change)	L8qq9PZyRg6ieKGEKhZolGC0vJWLw8iEJ88DRdyOg
Bearer token credentials	xvz1evFS4wEEPTGEFPHBog:L8qq9PZyRg6ieKGEKhZolGC0vJWLw8iEJ88DRdyOg
Base64 encoded bearer token credentials	eHZ6MWV2RlM0d0VFUFRHRUZQSEJvZzpMOHFxOVBaeVJnNmlIS0 dFS2hab2xHQzB2SldMdzhpRUo4OERSZHlPZw==

Submitting Keys

Example request (Authorization header has been wrapped):

```
POST /oauth2/token HTTP/1.1
Host: api.twitter.com
User-Agent: My Twitter App v1.0.23
Authorization: Basic eHZ6MWV2RlM0d0VFUFRHRUZQSEJvZzpMOHFxOVBaeVJn
                     NmllS0dFS2hab2xHQzB2SldMdzhpRUo4OERSZHlPZw==
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Content-Length: 29
Accept-Encoding: gzip

grant_type=client_credentials
```

If the request was formatted correctly, the server will respond with a JSON-encoded payload:

Example response:

```
HTTP/1.1 200 OK
Status: 200 OK
Content-Type: application/json; charset=utf-8
...
Content-Encoding: gzip
Content-Length: 140

{"token_type":"bearer","access_token":"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA%2FAAAA
AAAAAAAAAAAAAAAA%3DAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
```

Server: Routes

```ruby
require 'rubygems'
require 'bundler/setup'

require 'sinatra'
require 'pg'
require 'active_record'

require_relative 'db_config.rb'
require_relative 'ar_config.rb'
require_relative 'twitter_provider.rb'

before do
  response['Access-Control-Allow-Origin'] = '*'
end

def json_type
  { "Content-type" => "application/json" }
end

def search_for_tweets(app, query)
  twitter = TwitterProvider.new app
  resp = twitter.search(query, request["count"])
  [resp.code.to_i, {"Content-type" => resp.content_type }, [ resp.body ]]
end

def register_location_search(app, location)
  lat,lng,rad = location.split(",")
  Search.create :app => app,
                :lat => lat.to_f,
                :lng => lng.to_f,
                :radius => rad.gsub(/[^\d|.]/,"").to_f
end

get '/:appname/hashtag/:hashtag' do |appname, hashtag|
  app = App.find_by! name: appname
  Search.create :app => app, :hashtag => "#" + hashtag
  search_for_tweets app, URI::encode("q=-RT #" + hashtag)
end

get '/:appname/location/:location' do |appname, location|
  app = App.find_by! name: appname
  register_location_search app, location
  search_for_tweets app, "geocode=#{location}&q=-RT"
end
```

Server: Twitter

```ruby
class TwitterProvider
  # takes a credentials object that responds to:
  #  key, secret, to_s, save!
  def initialize(credentials)
    @app = credentials
    @api_version = 1.1
    @twitter_api = twitter_api
  end

  def search(query, count=20)
    twitter_get "search/tweets.json?#{query}&count=#{count}&result_type=recent&include_retweets=false"
  end

private
  def twitter_api
    api_url = URI "https://api.twitter.com"
    twitter = Net::HTTP.new api_url.host, api_url.port
    twitter.use_ssl = true
    twitter
  end

  def get_bearer
    if(!@app.bearer or @app.bearer.empty?)
      p "fetching bearer for #{@app}"
      credentials = Base64.encode64(URI::encode(@app.key) +
                                    ":" + URI::encode(@app.secret)).delete!("\n")
      auth_header = { "Authorization" => "Basic #{credentials}" }
      resp = @twitter_api.post '/oauth2/token', "grant_type=client_credentials", auth_header
      json = JSON.parse resp.body
      @app.bearer = json["access_token"]
      @app.save!
    end
    @app.bearer
  end

  def get_credentials_header
    p "getting credentials for #{@app}"
    { "Authorization" => "Bearer #{get_bearer()}" }
  end

  def twitter_get(method)
    path = format_call method
    p "twitter request(#{@app}): #{path}"
    @twitter_api.get path, get_credentials_header
  end

  def format_call(meth)
    "/#{@api_version}/#{meth}".gsub "//", "/"
  end
end
```

Trivia: Which Of the Following Is True?

A. I have taken Daniel Steinberg's iOS Precompiler 5 times

B. Saul Mora is speaking here next month

C. Apple has a KB Article Recommending Different Clothes and Lotions

D. I have an iPhone 6 With Me Today

E. I have a SpriteKit based game in the App Store

Answer: C

https://support.apple.com/kb/TS2729

Languages English

Apple Earbuds and static electricity

Symptoms

It's possible to receive a small and quick electrical (static) shock from your earbuds while listening to iPod, iPhone, or a Mac computer.

Resolution

What is happening?

When using headphones in areas where the air is very dry, it is easy to build up static electricity and possible for your ear to receive a small electrostatic discharge from the headphones. Receiving a static sho from a pair of earbuds does not necessarily indicate an issue with the iPod, iPhone, or earbuds.

This condition is very similar to dragging your feet across a carpet and receiving a static shock by touching a door knob. However, instead of the static charge building up on your body, the charge builds up on the device that the earbuds are connected to. Likewise, instead of the static buildup discharging through your finger when you touch a door knob, it discharges through the earbuds.

Note: This condition is not limited to Apple hardware. Static can potentially build up on almost any hardware and could be discharged using any brand of earbuds.

Why does static build up on my device?

Static buildup on your electronic device is largely due to the environment in which it is used. Certain environments and actions can encourage static electricity buildup.

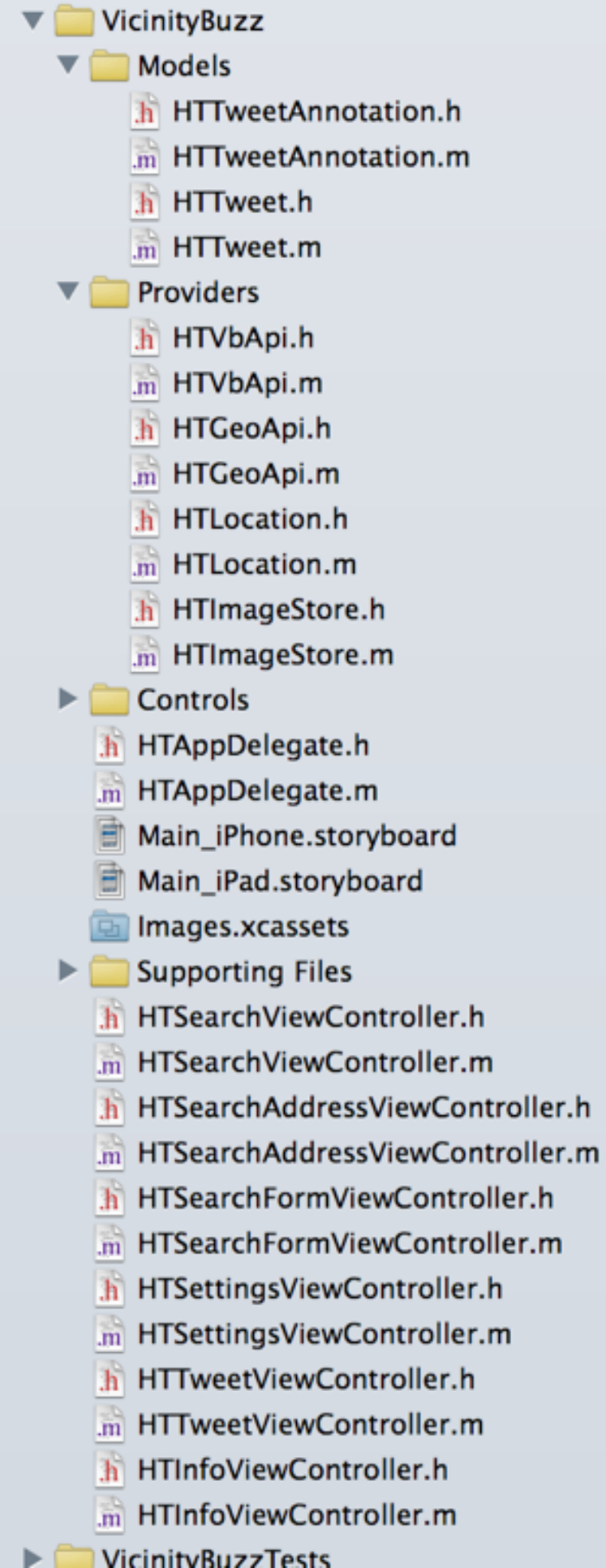
- Very low humidity (dry) environments are conducive for static buildup.
- Very windy environments are conducive for static buildup.
- Taking your device in and out of your pocket can create a static charge.
- Jogging or exercising with your device can cause a static charge.
- Clothes made with synthetic fibers (like nylon) can easily become charged with static electricity.

- If you have dry skin, try anti-static hand lotion.
- Try wearing different clothes. Try clothes with natural fibers since synthetic fibers are more likely to hold a static charge.

Indoors

- Try raising the moisture level in the air of the local environment by using a portable humidifier or adjusting the humidity control on your air conditioner.

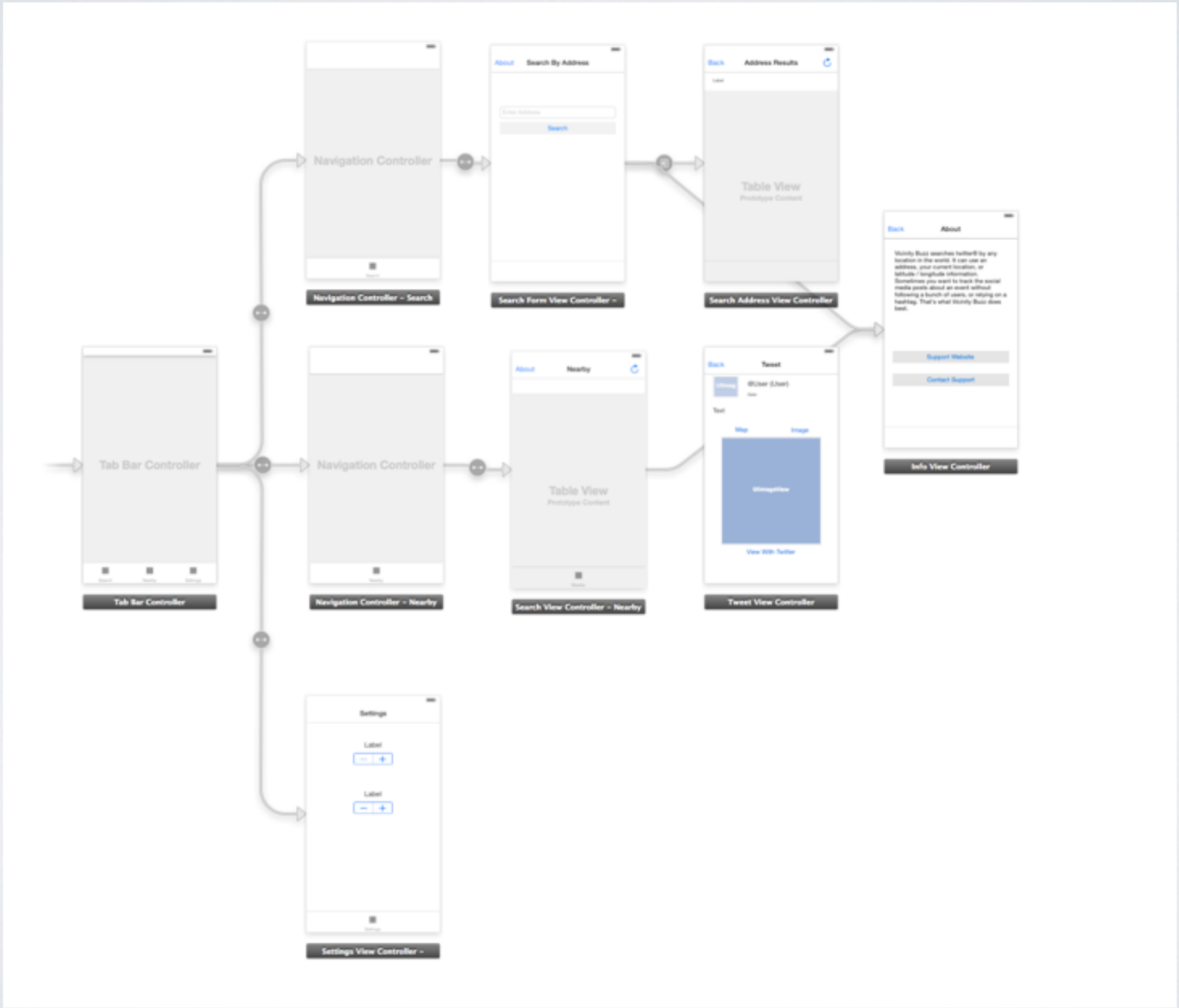
- There are a number of anti-static sprays that can be sprayed into the air that can be used to reduce static.



Structure

*copyright Universal Pictures

Storyboard


Navigation Controller
Navigation Controller – Search
About
Search By Address
Search Form View Controller –
Back
Address Results
Table View
Prototype Content
Search Address View Controller
Back
About
Support Website
Contact Support
Info View Controller
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Navigation Controller – Nearby
About
Nearby
Table View
Prototype Content
Search View Controller – Nearby
Back
Tweet
Text
Map
Image
View With Twitter
Tweet View Controller
Settings
Label
Label
Settings View Controller –

iOS Calling Server

```objc
-(void)searchWithLat:(double)lat
                Long:(double)lng
        WithCallback:(VbCallback)callback
{
    self.currentCallback = callback;
    NSLog(@"requesting data");
    NSString *searchAPI = @"http://api.vicinitybuzz.com/vbuzz-ios/location";
    NSString *uriWithParams = [NSString stringWithFormat:@"%@/%.4f,%.4f,%.1fmi?count=%d",
                               searchAPI, lat, lng, self.radius, self.postCount];
    NSString *encodedURI = [uriWithParams stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
    NSLog(@"sending GET query to: %@", encodedURI);
    NSURL *url = [NSURL URLWithString:encodedURI];
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];
    self.connection = [[NSURLConnection alloc] initWithRequest:request
                                                      delegate:self
                                              startImmediately:YES];
}

-(void)processResponse
{
    NSDictionary *json = [NSJSONSerialization JSONObjectWithData:self.jsonData
                                                         options:0
                                                           error:nil];
    NSArray *tweetsJson = [json objectForKey:@"statuses"];
    NSMutableArray *result = [NSMutableArray arrayWithCapacity:[tweetsJson count]];
    [tweetsJson enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL *stop) {
        [result addObject:[[HTTweet alloc] initWithDictionary:obj]];
    }];
    self.currentCallback([NSArray arrayWithArray:result]);
    NSLog(@"should load tweets");
}
```

Tweet

```
14  @interface HTTweet : NSObject
15
16  @property (nonatomic, strong) NSString *statusId;
17  @property (nonatomic, strong) NSString *realName;
18  @property (nonatomic, strong) NSString *screenName;
19  @property (nonatomic, strong) NSDate *createdAt;
20  @property (nonatomic, strong) NSString *text;
21  @property (nonatomic, strong) UIImage *profileImage;
22  @property (nonatomic, strong) UIImage *mediaImage;
23  @property (nonatomic) CLLocationCoordinate2D location;
24
25  -(id)initWithDictionary:(NSDictionary*)tweetJson;
26  -(NSString *)createdAtString;
27  -(NSString *)displayName;
28  -(BOOL)containsImage;
29  -(void)listenForImagesLoaded:(ImageLoadedCallback)callback;
```

Displaying In UITableView

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    HTCell *cell;

    if ([indexPath row] < [self.tweets count])
    {
        HTTweet *tweet = [self.tweets objectAtIndex:[indexPath row]];

        if ([tweet containsImage]) {
            cell = [tableView dequeueReusableCellWithIdentifier:@"vbMediaCell"];
            [cell.mediaImage setImage:[tweet mediaImage]];
        } else {
            cell = [tableView dequeueReusableCellWithIdentifier:@"vbCell"];
        }
        [cell.nameLabel setText:[tweet displayName]];
        [cell.dateLabel setText:[tweet createdAtString]];
        [cell.contentLabel setText:[tweet text]];
        [cell.profileImage setImage:[tweet profileImage]];
    } else {
        cell = [tableView dequeueReusableCellWithIdentifier:@"vbCell"];
        [cell.contentLabel setText:@"No more items!"];
    }
    return cell;
}

- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    HTTweet *tweet = [self.tweets objectAtIndex:[indexPath row]];
    if([tweet containsImage]) {
        return MEDIA_CELL_HEIGHT;
    } else {
        return CELL_HEIGHT;
    }
```

Current Location

```objc
#import "HTLocation.h"

@interface HTLocation ()

@property (nonatomic, strong) CLLocationManager *locationManager;
@property (copy) LocCallback currentCallback;

@end

@implementation HTLocation

-(id)init {
    self = [super init];
    if(self){
        self.locationManager = [[CLLocationManager alloc] init];
        self.locationManager.delegate = self;
        self.locationManager.distanceFilter = kCLDistanceFilterNone;
        self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
    }
    return self;
}

-(void)getCurrentLocationWithCallback:(LocCallback)callback
{
    self.currentCallback = callback;
    [self.locationManager startUpdatingLocation];
}

-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations
{
    [self.locationManager stopUpdatingLocation];
    self.currentCallback(locations[0]);
}

-(BOOL)hasPermission
{
    return [CLLocationManager locationServicesEnabled] &&
        !([CLLocationManager authorizationStatus] == kCLAuthorizationStatusDenied);
}
```

Geocoding

```
@interface HTGeoApi()

@property(copy) MapCallback currentCallback;
@property(strong, nonatomic) NSURLConnection *connection;
@property(strong, nonatomic) NSMutableData *jsonData;

@end

@implementation HTGeoApi

-(void)getAddressFromLat:(CLLocationDegrees)lat
                 andLong:(CLLocationDegrees)lng
            withCallback:(MapCallback)callback
{
    self.currentCallback = callback;
    NSString *query = [NSString stringWithFormat:@"latlng=%f,%f", lat, lng];
    [self requestFromGeoApi:query];
}

-(void)getLatLngFromAddress:(NSString *)address
                withCallback:(MapCallback)callback
{
    self.currentCallback = callback;
    NSString *query = [NSString stringWithFormat:@"address=%@", address];
    [self requestFromGeoApi:query];
}

-(void)requestFromGeoApi:(NSString*)query
{
    NSString *mapAPI = @"http://maps.googleapis.com/maps/api/geocode/json?sensor=false&";
    NSString *uriWithParams = [mapAPI stringByAppendingString:query];
    NSString *encodedURI = [uriWithParams stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
    NSLog(@"sending GET query to: %@", encodedURI);
    NSURL *url = [NSURL URLWithString:encodedURI];
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];
    self.connection = [[NSURLConnection alloc] initWithRequest:request
                                                      delegate:self
                                              startImmediately:YES];
}

-(void)processResponse
```

Map Display

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.userLabel.text = self.tweet.displayName;
    self.dateLabel.text = self.tweet.createdAtString;
    self.textLabel.text = self.tweet.text;
    self.profileImageView.image = self.tweet.profileImage;
    self.mediaImageView.image = self.tweet.mediaImage;
    self.mapView.delegate = self;
    HTTweetAnnotation *tweetAnnotation = [[HTTweetAnnotation alloc]
                                          initWithCoordinate:self.tweet.location
                                                       title:[@"@" stringByAppendingString:self.tweet.screenName]
                                                 andSubTitle:self.tweet.createdAtString];
    [self.mapView addAnnotation:tweetAnnotation];
    HTTweetAnnotation *searchAnnotation = [[HTTweetAnnotation alloc]
                                           initWithCoordinate:self.searchLocation
                                                        title:@"Search Center"
                                                  andSubTitle:@""];
    [self.mapView addAnnotation:searchAnnotation];
    NSNumber *radius = [[NSUserDefaults standardUserDefaults] objectForKey:RADIUS_KEY];
    float meters = ([radius floatValue] + 1.0) * 2 * 1609;
    MKCoordinateRegion region = MKCoordinateRegionMakeWithDistance(self.searchLocation, meters, meters);
    [self.mapView setRegion:region animated:YES];

    if(![self.tweet containsImage]) {
        [self.mediaImageView setHidden:YES];
        [self.mapButton setHidden:YES];
        [self.mediaButton setHidden:YES];
    }
    [self viewMap:nil];
}
```

Map Annotation

```objc
-(MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id<MKAnnotation>)annotation
{
    if([annotation isKindOfClass:[HTTweetAnnotation class]]) {
        MKPinAnnotationView *aView = (MKPinAnnotationView *)[self.mapView
            dequeueReusableAnnotationViewWithIdentifier:@"tweet"];
        if(!aView) {
            aView = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                                     reuseIdentifier:@"tweet"];
            if([[annotation title] isEqualToString:@"Search Center"]) {
                aView.pinColor = MKPinAnnotationColorRed;
            } else {
                aView.pinColor = MKPinAnnotationColorGreen;
            }
            aView.animatesDrop = YES;
            aView.canShowCallout = YES;
        } else {
            aView.annotation = annotation;
        }
        return aView;
    }
    return nil;
}
```

Image Storage Singleton

```objc
@implementation HTImageStore

+(id)getStore
{
    static HTImageStore* _singleton = nil;
    if (_singleton == nil) {
        _singleton = [[HTImageStore alloc] init];
    }
    return _singleton;
}

-(id)init
{
    self = [super init];
    if(self) {
        self.imageCache = [[NSMutableDictionary alloc] init];
    }
    return self;
}

-(void)getImageFromURL:(NSString *)address withBlock:(ImageCallback)callback
{
    if([[self.imageCache allKeys] containsObject:address]) {
        return callback([self.imageCache objectForKey:address]);
    } else {
        NSURL *url = [NSURL URLWithString:address];
        [NSURLConnection sendAsynchronousRequest:[NSURLRequest requestWithURL:url]
                                           queue:[NSOperationQueue mainQueue]
                               completionHandler:^(NSURLResponse *response, NSData *data, NSError *connectionError)
                                   {
                                   UIImage *img = [UIImage imageWithData:data];
                                   [self.imageCache setObject:img forKey:address];
                                   callback(img);
                               }];
    }
}

@end
```

Inheritance…

```
9  #import "HTSearchAddressViewController.h"
10
11 @implementation HTSearchAddressViewController
12
13 -(IBAction)goBack:(id)sender
14 {
15     [self.navigationController popToRootViewControllerAnimated:YES];
16 }
17
18 -(void)defaultSearch
19 {
20     [self searchByAddress];
21 }
22
23 @end
```

Challenges

- Non-flow Layout
    - Custom Cells
- User Experience / Flow of Application
- Table View Inset
- Images need downloading, not just URL
    - complicated callbacks
- Date formats
- Map Coordinates Backward…
- Settings Classes (Is NSDefaults correct?)
- Code Structure in Objective-C
    - Different but not…
- Strong vs Weak, Property vs iVar

Things I Liked About Native & Objective-C

- Responsiveness
- Blocks
    - typedef for callback
- APIs for Email, Safari, Twitter Interaction
- Inherited a ViewController…
    - Good? Bad?
- Storyboards
    - Once I got used to them
- New Syntax Shortcuts
    - @import…, arrays…

Coming/Potential Features

- Bookmark Locations
- Suggested Locations
- Trending Locations
- White Labeling

Finis

- Questions?

- Suggestions?