

MapKit Annotations

Fetching from GeoCouchDB and Core Data

Matt Blair

Overview

Adding annotations based on region:

- From a remote-document database
- From Core Data on the device

(Mostly) Similar Things

SQL	Database	Table	Row	Field
Objective-C		Class	Instance	Ivars, Properties
JSON		Array	Object	Property?
CouchDB	Database	Document Type (optional)	Document	key/value, field
Core Data	Datastore	Entity	Instance	Attribute

Geoquery

“Show me everything in this particular view of the map.”

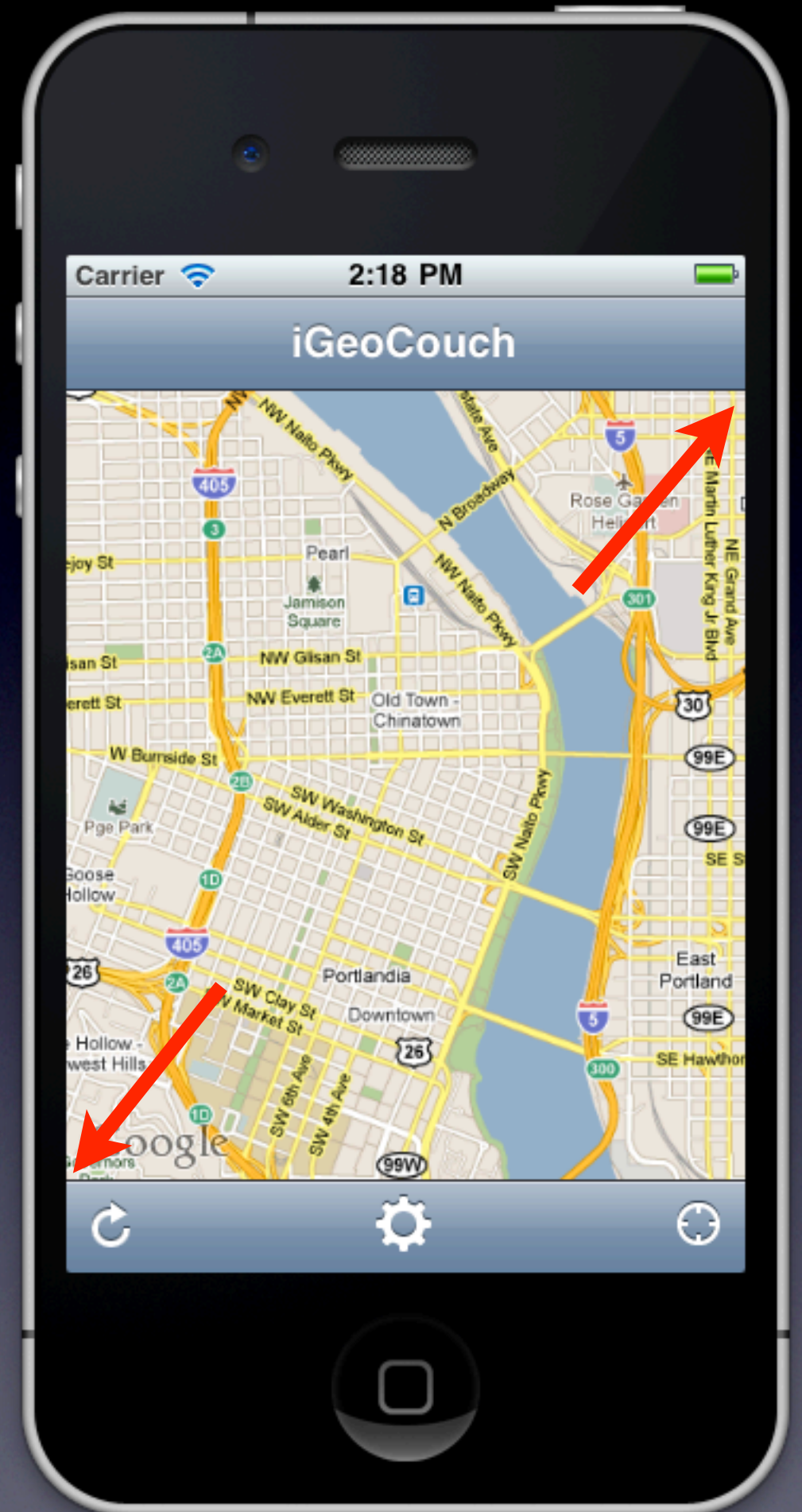
Fetching From a Remote Source

What is CouchDB?

- Document Database (NoSQL to some)
- Everything stored as JSON (including code)
- HTTP-based API (GET/POST, etc.)
- GeoCouch extension supports geoqueries
- Example: PDXAPI.com
- Hosting: CouchOne.com

Bounding Box Queries

- All documents in a rectangular area
- Defined by SW and NE corners



Bounding Box Query

[http://elsewise.couchone.com/
fire_sta/_design/igeocouch/
_spatial/points?
bbox=-122.689733505249,45.5051
4553351729,-122.6622676849365,
45.53280958598629](http://elsewise.couchone.com/fire_sta/_design/igeocouch/_spatial/points?bbox=-122.689733505249,45.50514553351729,-122.6622676849365,45.53280958598629)

Query Components

[http://elsewise.couchone.com/
fire_sta/_design/igeocouch/
_spatial/points?](http://elsewise.couchone.com/fire_sta/_design/igeocouch/_spatial/points?bbox=-122.689733505249,45.50514553351729,-122.6622676849365,45.53280958598629)

[bbox=-122.689733505249,45.5051
4553351729,-122.6622676849365,
45.53280958598629](http://elsewise.couchone.com/fire_sta/_design/igeocouch/_spatial/points?bbox=-122.689733505249,45.50514553351729,-122.6622676849365,45.53280958598629)

bbox parameter

```
bbox=
```

```
// SW
```

```
-122.6785969, 45.5167974,
```

```
// NE
```

```
-122.6763439, 45.51772964
```


Problem

MKCoordinateRegions are defined by a center point and span deltas, not a bounding box.

MKCoordinateRegion

- center (CLLocationCoordinate2D)
- span (MKCoordinateSpan)

MK or CL?

MapKit and CoreLocation intermingle. A lot.

region.center

- latitude (CLLocationDegrees)
- longitude (CLLocationDegrees)

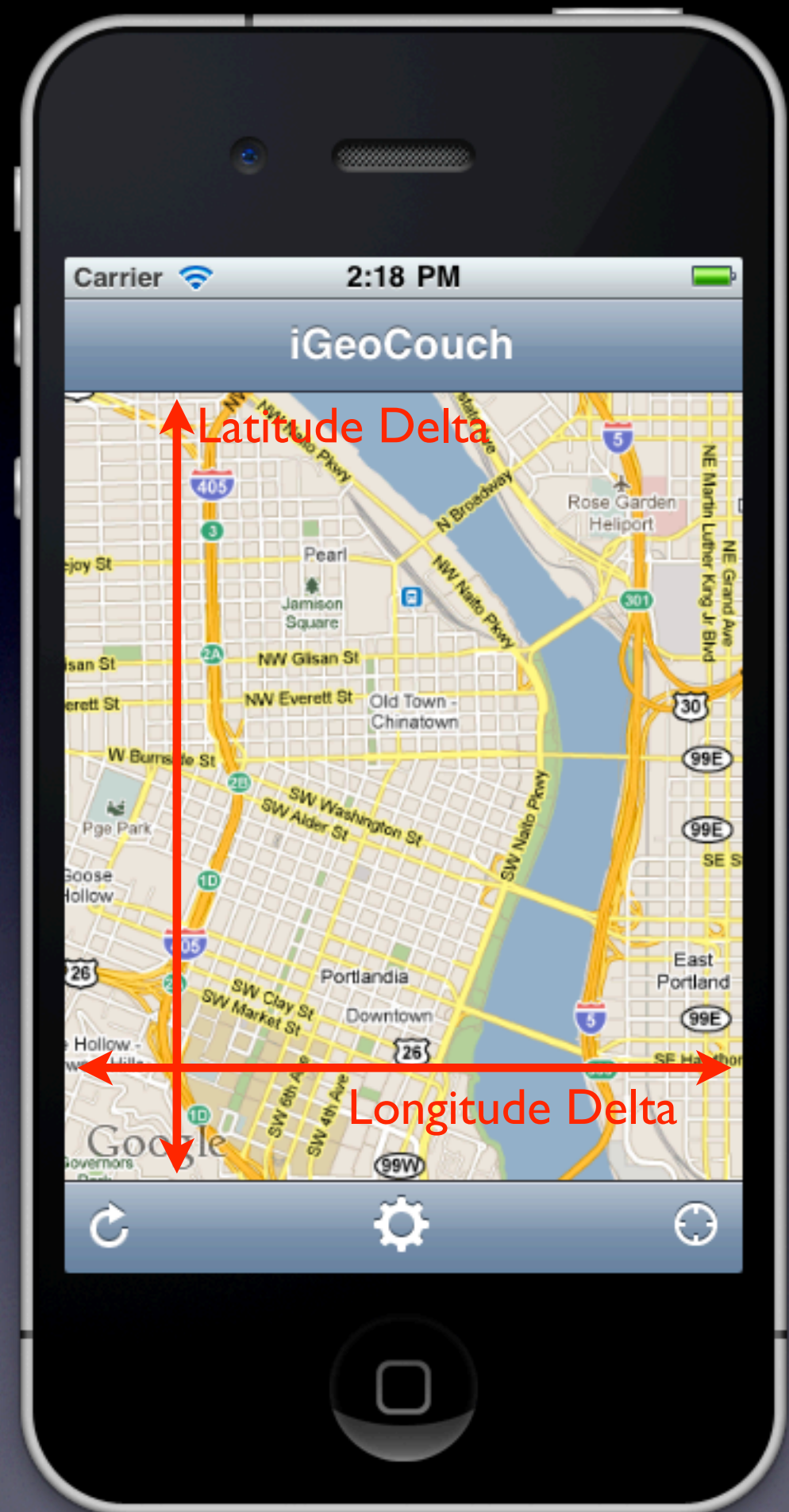
CLLocationDegrees?

```
typedef double CLLocationDegrees;
```


region.span

- latitudeDelta (CLLocationDegrees)
- longitudeDelta (CLLocationDegrees)

The Deltas



Region to Bounding Box

```
MKCoordinateRegion region = theMapView.region;
```

```
NSNumber *southLatitude = [NSNumber  
numberWithDouble:region.center.latitude -  
region.span.latitudeDelta/2.0];
```

```
NSNumber *northLatitude = [NSNumber  
numberWithDouble:region.center.latitude +  
region.span.latitudeDelta/2.0];
```

```
NSNumber *westLongitude = [NSNumber  
numberWithDouble:region.center.longitude -  
region.span.longitudeDelta/2.0];
```

```
NSNumber *eastLongitude = [NSNumber  
numberWithDouble:region.center.longitude +  
region.span.longitudeDelta/2.0];
```


Construct the URL

```
NSString *urlString = [[NSString alloc]
initWithFormat:@"%d%%d%%d,%d,%d,%d",
kDatabaseURL, kPathForMapSearch,
westLongitude, southLatitude,
eastLongitude, northLatitude];
```


Construct the URL (without constants)

```
NSString *urlString = [[NSString alloc]
initWithFormat:@"http://elsewise.couchone.com/
fire_sta/_design/igeocouch/_spatial/points?bbox=%@,
%@, %@, %@, %@, %@",
westLongitude, southLatitude,
eastLongitude, northLatitude];
```


Request the Data

- ASIHTTPRequest
- json.framework

Results are JSON

```
{ "rows":  
  [{ "id": "df5d405a5afaba4b77b89cfd7e3320d7", "bbox":  
    [-122.682959988765, 45.5097995643119, -122.682959988765,  
    45.5097995643119], "value": { "title": "Fire Station  
4", "subtitle": "511 SW COLLEGE ST" } },  
  
    { "id": "df5d405a5afaba4b77b89cfd7e333036", "bbox":  
    [-122.688716309701, 45.5287977374666, -122.688716309701,  
    45.5287977374666], "value": { "title": "Fire Station  
3", "subtitle": "1715 NW JOHNSON ST" } },  
  
    { "id": "df5d405a5afaba4b77b89cfd7e31fd18", "bbox":  
    [-122.670704990545, 45.5218177487472, -122.670704990545,  
    45.5218177487472], "value": { "title": "Fire Station  
1", "subtitle": "55 SW ASH ST" } }  
  
  ] }
```


Pretty Printing

- Copy chunk of JSON output to pasteboard
- In Terminal:

```
pbpaste | python -mjson.tool | pbcopy
```


A Fire Station

```
{  
  "bbox": [  
    -122.670704990545,  
    45.521817748747203,  
    -122.670704990545,  
    45.521817748747203  
  ],  
  "id": "df5d405a5afaba4b77b89cfd7e31fd18",  
  "value": {  
    "subtitle": "55 SW ASH ST",  
    "title": "Fire Station 1"  
  }  
}
```


We Need Annotations

- `json.framework` gives us an array of `NSDictionaries`
- Convert to an array of objects that conform to `MKAnnotation` protocol

MKAnnotation Protocol

```
@interface GeoCouchAnnotation : NSObject <MKAnnotation> {  
    NSString *title;  
    NSString *subtitle;  
    NSNumber *latitude;  
    NSNumber *longitude;  
    CLLocationCoordinate2D coordinate; //required  
    NSString *pointID; // not shown or required  
    // can declare other ivars here, too...  
}
```


Simple Getter Method

```
- (CLLocationCoordinate2D) coordinate {  
    coordinate.latitude = [self.latitude doubleValue];  
    coordinate.longitude = [self.longitude doubleValue];  
    return coordinate;  
}
```


JSON to Annotation

```
for (NSDictionary *aPoint in pointsArrayFromJSON) {  
  
    ...  
  
    [ga setTitle:[aPoint valueForKey:@"value"]  
valueForKey:@"title"]];  
  
    [ga setSubtitle:[aPoint valueForKey:@"value"]  
valueForKey:@"subtitle"]];  
  
    [ga setLatitude:[NSNumber numberWithDouble:[aPoint  
valueForKey:@"bbox"] objectAtIndex:1] doubleValue]]];  
  
    [ga setLongitude:[NSNumber numberWithDouble:[aPoint  
valueForKey:@"bbox"] objectAtIndex:0] doubleValue]]];  
  
    [ga setPointID:[aPoint valueForKey:@"id"]];  
  
    ...  
}
```


Adding Annotations to your MKMapView

- Get array of current annotations
- Remove them
- Add array of new annotations

Adding Annotations

```
NSArray *oldAnnotations = theMapView.annotations;
```

```
[theMapView removeAnnotations:oldAnnotations];
```

```
[theMapView addAnnotations:pointAnnotationArray];
```


Keep Current Location

It can take a while to come back.

Keep Current Location

```
NSPredicate *userLocationPredicate =  
[NSPredicate predicateWithFormat:  
@"!(self isKindOfClass: %@)",  
[MKUserLocation class]];
```


Keep Current Location

```
NSArray *oldAnnotations = theMapView.annotations;

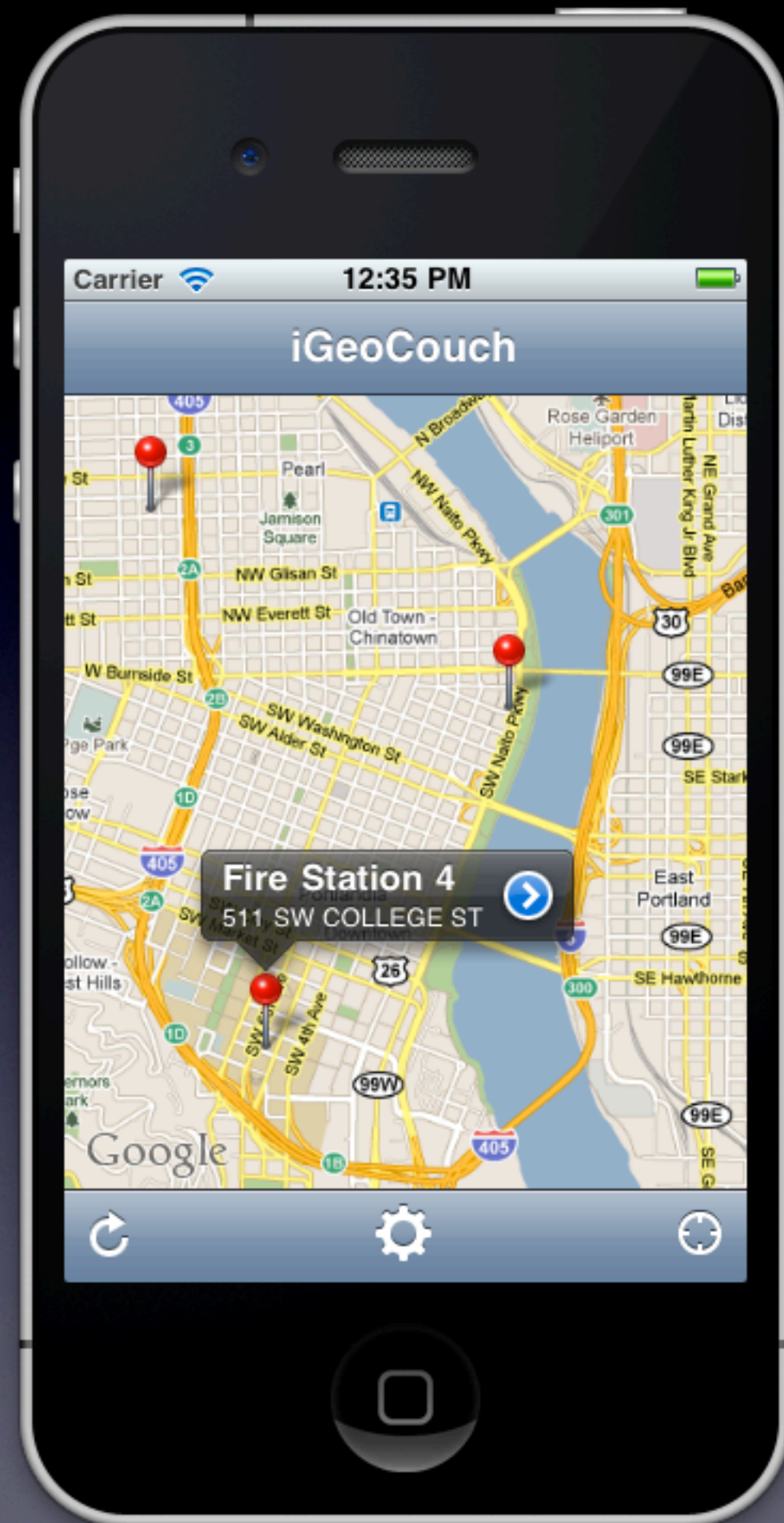
NSPredicate *userLocationPredicate = [NSPredicate
predicateWithFormat:@"!(self isKindOfClass: %@)",
[MKUserLocation class]];

NSArray *annotationsToRemove = [oldAnnotations
filteredArrayUsingPredicate:userLocationPredicate];

[theMapView removeAnnotations:annotationsToRemove];

[theMapView addAnnotations:pointAnnotationArray];
```


Finally!



How did we get that callout?

Customizing Annotation Views

//MKMapView delegate method

– mapView:viewForAnnotation:

Customizing Annotation Views

...

```
pinView.pinColor = MKPinAnnotationColorPurple;
```

```
pinView.animatesDrop = YES; // MKPinAnnotationView only
```

```
pinView.canShowCallout = YES;
```

```
UIButton* rightButton = UIButtonbuttonWithType:
```

```
    UIButtonTypeDetailDisclosure];
```

```
pinView.rightCalloutAccessoryView = rightButton;
```

...

Showing Details

```
// add target/action to button in callout
```

```
// ...or use this MKMapView delegate method
```

```
- (void)mapView:(MKMapView *)mapView annotationView:  
(MKAnnotationView *)view calloutAccessoryControlTapped:  
(UIControl *)control
```


Showing Details

```
- (void)mapView:(MKMapView *)mapView annotationView:  
(MKAnnotationView *)view calloutAccessoryControlTapped:  
(UIControl *)control
```


Access the Annotation

```
GeoCouchAnnotation *selectedPoint = view.annotation;

PointDetailViewController *pdvc =
[[PointDetailViewController alloc]
initWithNibName:@"PointDetailViewController" bundle:nil];

pdvc.pointID = [selectedPoint pointID];

// push pdvc onto navigation controller

// pdvc fetches and displays details in its viewDidLoad
```


Remote = Good

- Browse arbitrary geo-datasets
- Edits to central dataset instantly available to all users
- Highly scalable

Remote != Good

- `_id`, `_rev`, can require multiple requests

Key ▼	Value
"b916e524b2e1f24e72ac7a81aa5a4925" ID: b916e524b2e1f24e72ac7a81aa5a4925	{rev: "5-2d9826d9d3824946377900b3514cd777"}
"b916e524b2e1f24e72ac7a81aa5a41b4" ID: b916e524b2e1f24e72ac7a81aa5a41b4	{rev: "5-49e578738a1eb1c2d4a940f8006f156b"}
"b916e524b2e1f24e72ac7a81aa5a39a9" ID: b916e524b2e1f24e72ac7a81aa5a39a9	{rev: "5-c8ca20f7199f774e1c99fa4afdf8dc9e"}
"b916e524b2e1f24e72ac7a81aa5a3044" ID: b916e524b2e1f24e72ac7a81aa5a3044	{rev: "5-3b8a1280eb795b178a75f67913c24c0b"}
"b916e524b2e1f24e72ac7a81aa5a212c"	{rev: "5-6b41d0ada01ca6f80bc547dedb9252bf"}

Remote != Good

- How do you display ad-hoc/mutable data structures beautifully?

Remote != Good

Over-the-air performance is:

- variable
- uncontrollable
- at times, unusable

Using Core Data

SQLite is one option for datastores

DIY SQLite != Future-Proof

By DIY, I mean:

- dropping in your own database
- hacking the db directly
- building an sqlite file in another language

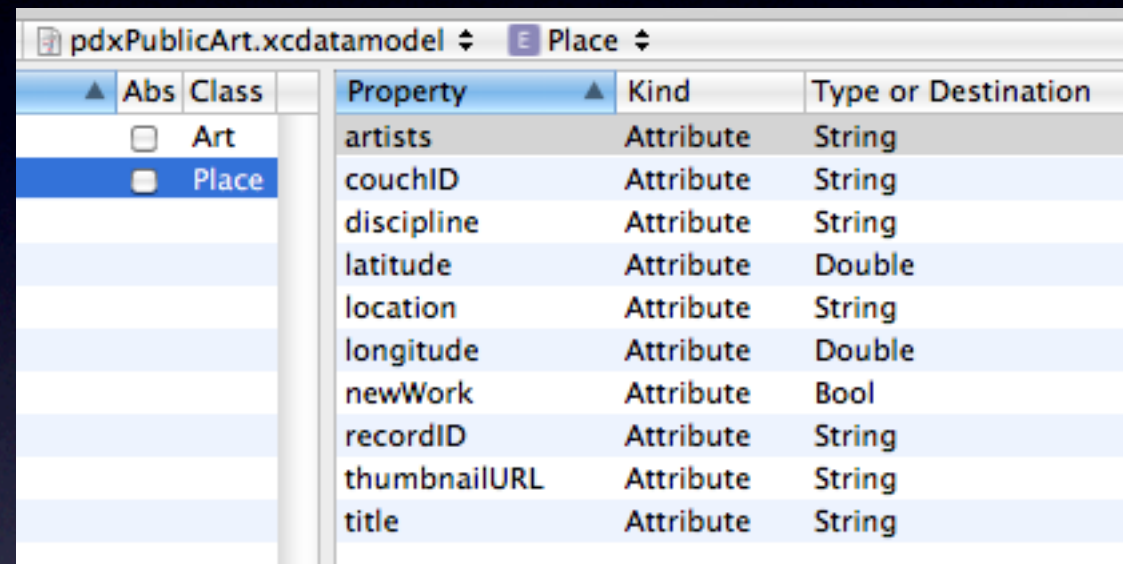
Apple's Refrain

“Don't do that.”

Core Data Import

- Code in Objective-C
- iOS and OS X can share Core Data Models

Storing Coordinates in Core Data



The screenshot shows the Xcode interface for editing a Core Data model. The title bar indicates the file is 'pdxPublicArt.xcdatamodel' and the selected entity is 'Place'. On the left, a sidebar shows the 'Class' tab with 'Art' and 'Place' listed. The main area displays a table of attributes for the 'Place' class.

Property	Kind	Type or Destination
artists	Attribute	String
couchID	Attribute	String
discipline	Attribute	String
latitude	Attribute	Double
location	Attribute	String
longitude	Attribute	Double
newWork	Attribute	Bool
recordID	Attribute	String
thumbnailURL	Attribute	String
title	Attribute	String

- Coordinates are doubles in model
- Latitude and Longitude are NSNumbers in Class definition

CLLocationDegrees

```
typedef struct {  
    CLLocationDegrees latitude;  
    CLLocationDegrees longitude;  
} CLLocationCoordinate2D;  
  
typedef double CLLocationDegrees;
```


Conversions

```
//NSNumber to double
```

```
CLLocationCoordinate2D originalLocation;
```

```
originalLocation.latitude = [[theArt latitude] doubleValue];
```

• • •

```
//double to NSNumber
```

```
[self setNewLatitude:[NSNumber  
numberWithDouble:theAnnotation.coordinate.latitude]];
```

• • •

Core Data Geoqueries

```
// fetch setup boilerplate

NSFetchRequest *fetchRequest = [[NSFetchRequest
alloc] init];

[fetchRequest setEntity:[NSEntityDescription
entityWithName:@"Place"
inManagedObjectContext:self.managedObjectContext]];
```


Core Data Geoqueries

```
// calculate bounding box from region as before

NSPredicate *predicate;

predicate = [NSPredicate predicateWithFormat:

@"latitude>%@", AND latitude<%@", AND longitude>%@", AND
longitude<%@",

latitudeSouth, latitudeNorth,

longitudeWest, longitudeEast];

[fetchRequest setPredicate:predicate];
```


Core Data Geoqueries

```
// (continued)
```

```
// add sort descriptors
```

```
NSError *error = nil;
```

```
NSArray *fetchedPlaces = [self.managedObjectContext  
executeFetchRequest:fetchRequest error:&error];
```

```
// now we have an array of Place objects
```


Array to Array

- Core Data returns an array of our `NSManagedObject` subclass
- Convert to an array of objects that adopt `MKAnnotation` protocol, just like we did with JSON

MKAnnotation Protocol

- Can NSManagedObject subclasses adopt MKAnnotation Protocol directly?
- Transient Attributes (maybe)

Use Case #1: Public Art

- Not 1-to-1 relationship between Art and Place
- Searching Art, but mapping Place
- e.g. 40 works of art at 12 places
- 40 Art objects, 12 annotations

Use Case #2: Zoom-based Clustering

Or any kind of
processing/
aggregation of
objects into
annotations



(ushahidi.com)

Geo + Wildcard Search

```
// wildcard at specific coordinates  
  
predicate = [NSPredicatepredicateWithFormat:  
  
@"latitude=%@ AND longitude=%@ AND ((title contains  
[cd] %@) OR (artists contains[cd] %@))",  
  
self.latitude, self.longitude,  
self.currentSearchString, self.currentSearchString];
```

From *Core Data Programming Guide*:

"If the predicate combines textual and non-textual comparisons, then it is likely to be more efficient to specify the non-textual predicates first..."

Local = Good

Fast!

Fewer over-the-air dependencies...

Known Datastructures Allow Better Design

Heritage Trees

Endlicher pine


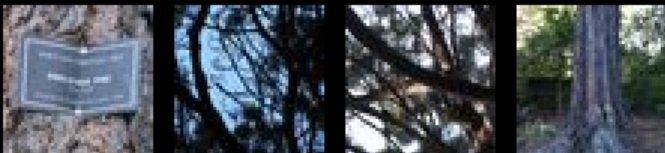
i *Pinus rudis*

2403 SW JEFFERSON ST

SW Rose Garden Blvd, Washington
Park (SE end parking lot) Rare

Height: 60 feet
Circumference: 8.1 feet
Diameter: 31.0 inches
Spread: 37 feet

Designated in 1999



Off-line Use?

- Use case: preview an area in the classroom with iPod Touch devices
- Neighborhood-level/Field-trip Quality
- Map tile cache cleared by memory warnings and power cycles

Local != Great

- Highly dynamic datasets
- Syncing to data in the cloud
- Dynamic data structures and metadata migrations

Thank You

Matt Blair

<http://elsewisestrategic.com>

- <http://pdxtrees.org>