Mobile Computing and the Cloud

Lecture 10

Jonathan R. Engelsma

CLOUD COMPUTING CATEGORIES

- Infrastructure as a Service (IaaS)
  - deliver computer infrastructure as a service
- Platform as a Service(PaaS)
  - deliver infrastructure and a software stack as a service
- Software as a Service (SaaS)
  - deliver an application as a service

NEW CATEGORY!

- Mobile Backend as a Service (MBaas)
  - parse.com
  - kinvey.com
  - firebase.com
  - kii.com

MBAAS

- Takes the complexity out of building a backend platform for a mobile app.
- Really just a subset of PaaS.
- Driven in part by "Mobile First" type deployments

PARSE.COM

- We will focus on parse.com
- Acquired by Facebook in April 2013!
- Provides
  - login / authentication services
  - user management
  - easy to create RESTful APIs
  - push notifications

Parse

SETTING UP PARSE (1)

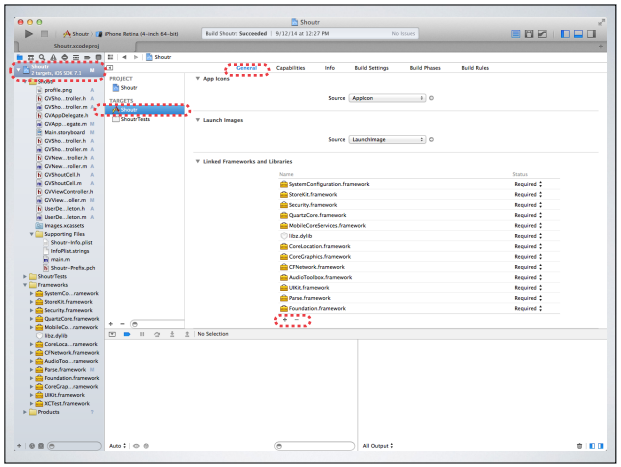
Assumption: Adding to an existing project.

- Step 1: Download the parse.com iOS framework:
  - https://parse.com/docs/downloads
- Step 2: Unpack and drag the Parse.framework, ParseUI.framework and Bolts.framework to your project folders in Xcode (copy!)

Parse

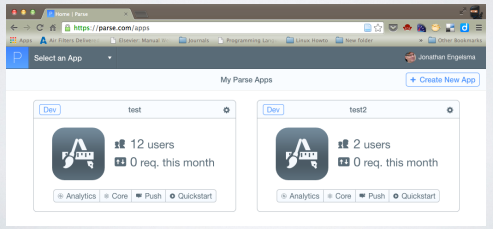
SETTING UP PARSE(2)

- Step 3: Manually add the following dependencies:
  - AudioToolbox.framework
  - CFNetwork.framework
  - CoreGraphics.framework
  - CoreLocation.framework
  - MobileCoreServices.framework
  - Accounts.framework
  - QuartzCore.framework
  - Security.framework
  - StoreKit.framework
  - SystemConfiguration.framework
  - Social.framework
  - libz.dylib & libsqlite3.dylib



SETTING UP PARSE(3)

- Step 4: Register for a free account on parse.com and establish a new app instance.



SETTING UP PARSE (4)

- Step 5: Point app to appropriate Parse app instance in AppDelegate implementation.

```
import Parse
import Bolts

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

  func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject:
    // [Optional] Power your app with Local Datastore. For more info, go to
    // https://parse.com/docs/ios_guide#localdatastore/iOS
    Parse.enableLocalDatastore()

    // Initialize Parse.
    Parse.setApplicationId("WpwqyrP8rszNK9rSQLuQTEZr8VRXO9Ek71y8oh3U",
      clientKey: "q5vLq9zEH5E0lGwVdNiZzrhktJmPRqnUQrHpn6dy")

    // [Optional] Track statistics around application opens.
    PFAnalytics.trackAppOpenedWithLaunchOptions(launchOptions)

    // ...
}

// ...
```

SETTING UP PARSE (5)

- Step 6: Add code to controller to persist objects:

a) import Parse at the top of your Swift source file

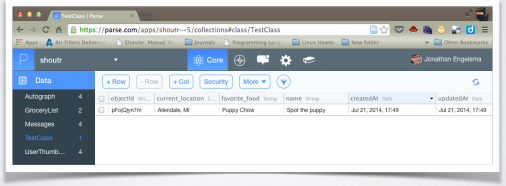
```
import Parse
```

b) add code to persist data from parse where appropriate.

```
let testObject = PFObject(className: "TestObject")
testObject["foo"] = "bar"
testObject.saveInBackgroundWithBlock { (success: Bool, error: NSError?) -> Void in
  println("Object has been saved.")
}
```

SETTING UP PARSE (6)

- Step 7: Validate the data has been persisted on parse.com:



SETTING UP PARSE (7)

- Step 8: Add code that queries for previously persisted objects!

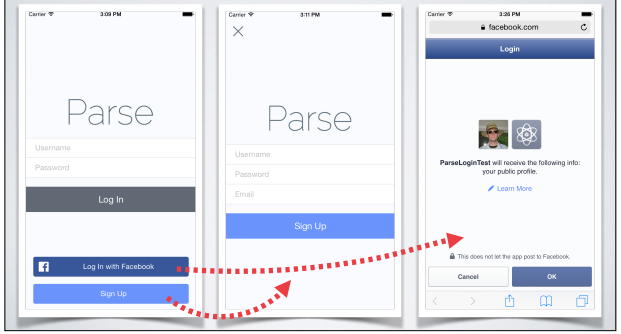
```
PFQuery *query = [PFQuery queryWithClassName:@"GameScore"];
[query whereKey:@"playerName" equalTo:@"Dan Stemkoski"];
[query findObjectsInBackgroundWithBlock:^(NSArray *objects, NSError *error) {
  if (!error) {
    // The find succeeded.
    NSLog(@"Successfully retrieved %d scores.", objects.count);
    // Do something with the found objects
    for (PFObject *object in objects) {
        NSLog(@"%@", object.objectId);
    }
  } else {
    // Log details of the failure
    NSLog(@"Error: %@ %@", error, [error userInfo]);
  }
}];
```

Objective-C Swift

SAVING/RETRIEVING DATA FROM PARSE.COM

Demo!

MANAGING USERS WITH PARSE.COM



MANAGING USERS (1)

- Preliminaries:
  - Add parse.com ParseUI.framework and ParseFacebookUtilsV4.framework to Xcode.
  - Download FB iOS SDK and add framework and dependencies to your Xcode project.
  - Add FB SDK headers to your bridging header in order to have visibility in Swift.

MANAGING USERS (2)

- On the network side:
  - Make sure you have a parse app instance created for your iOS app.
  - Make sure you have created a Facebook app instances created for your iOS app.

MANAGING USERS (3)

- AppDelegate mods

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject:
    AnyObject]?) -> Bool {

    Parse.enableLocalDatastore()
    Parse.setApplicationId("3CBekwGUL5CMlPPEqX7tmehm4sjlRgrDUtzITuec", clientKey:
        "rICIfyow66ak7ApyxfMwvzO89OK10nnZmZwhy12J")
    PFAnalytics.trackAppOpenedWithLaunchOptionsInBackground(launchOptions, block: nil)

    PFFacebookUtils.initializeFacebookWithApplicationLaunchOptions(launchOptions)
    return true
}
```

```
func applicationDidBecomeActive(application: UIApplication) {
    FBSDKAppEvents.activateApp()
}

func application(application: UIApplication, openURL url: NSURL, sourceApplication: String?,
    annotation: AnyObject?) -> Bool
{
    return FBSDKApplicationDelegate.sharedInstance().application(application, openURL: url,
        sourceApplication: sourceApplication, annotation: annotation)
}
```

MANAGE USERS (4)

- ViewController mods:

```
override func viewDidAppear(animated: Bool) {
    super.viewDidAppear(animated)
    if PFUser.currentUser() == nil {
        var loginCtrl = PFLogInViewController()
        loginCtrl.fields = PFLogInFields.UsernameAndPassword | PFLogInFields.LogInButton |
            PFLogInFields.Facebook | PFLogInFields.SignUpButton
        loginCtrl.facebookPermissions = ["friends_about_me"]
        loginCtrl.delegate = self

        var signupCtrl = PFSignUpViewController()
        signupCtrl.delegate = self
        loginCtrl.signUpController = signupCtrl

        self.presentViewController(loginCtrl, animated: true, completion: nil)
    } else {
        self.loginMessage.text = PFUser.currentUser()?.username
    }

}
```

MANAGING USERS (5)

- View Controller implements the delegates for PFLogInViewController and PFSignUpViewController:

```
class ViewController: UIViewController,
    PFLogInViewControllerDelegate, PFSignUpViewControllerDelegate
```

  - methods for
    - logging in, canceling, errors, validation, etc.

MANAGING USERS WITH PARSE.COM

Demo!