

# iOS Dev Accelerator

## Week 6 Day 2

- OAuth

# OAuth

- OAuth is an authentication protocol that allows two apps to interact and share resources.
- There are 3 actors in the OAuth workflow:
  - The service provider: Ex: Twitter
  - The consumer: Ex: Our app
  - The user
- The main benefit is that the user never shares his account and password with the consumer app.

# OAUTH Workflow

- Step 1: The user shows intent by attempting to post from the consumer app to the service provider(ex twitter)
- Step 2: Our app (The consumer) redirects the user to the service provider for authentication
- Step 3:The user gives permission to the service provider for all the actions the consumer should be able to do on his behalf (ex: posting to his timeline, accessing his twitter photos, etc)
- Step 4:The service provider returns the user to the consumer, with a request token
- Step 5:The consumer now sends the request token, together with its secret key to the service provider in exchange for an authentication token
- Step 6: The user performs actions and passes the authentication token with each call to prove who he is

# Callback URI

- The callback URI is just the callback entry point of your app.
- The service provider calls this URI to get the user back to the consumer app.
- All apps can be launched from either another app or from the browser itself.

# Registering a custom URL scheme

- Go to your info.plist and choose Add Row
- Select URL types from the drop down (its an array with one dictionary)
- In the item 0 dictionary, add an entry for the key 'URL Identifier'
- This entry will be the name of the custom url scheme you are defining. It is recommended to ensure uniqueness that you use reverse dns 'com.yourCompany.yourApp'
- Add another entry from the drop down called 'URL Schemes' which is an array
- For Item 0, give it the string you want to be your URL Scheme. For example, if you type MyApp. The custom URL for your app will be MyApp://

# Try it out!

- Launch your app after editing your plist with your URL scheme.
- Now launch Safari from your simulator, and enter in your app's URL into the URL bar.
- Your app should now open.

# Parsing(-vick) information that is passed with these URL's

- Often times, and especially with OAuth, your app's url will be called with extra parameters. Typically this is a token or flag.
- There is a method you can implement in your app delegate to intercept these URL calls:

```
- (BOOL)application:(UIApplication *)application  
    openURL:(NSURL *)url  
    sourceApplication:(NSString *)sourceApplication  
    annotation:(id)annotation
```