iOS Dev Accelerator
Week 3 Day 3

- JSON
- Adaptive Design and UISplitViewControllers
- Dynamic Tableview Size

JSON

- “JavaScript Object Notation” (but it really is language independent)
- “open standard format that uses human readable text to transmit data objects consisting of attribute-value pairs”
- Used primarily for communication between the server and client
- Is a more popular alternative to XML
- The official internet media type of JSON is application/json

JSON data types

- Number : Decimal number that makes no distinction between an integer and float
- String : a sequence of zero or more unicode characters. Delimited with double quotation marks, and escaped with a backslash
- Boolean : true or false vales
- Array: An ordered list of zero or more values, can be of any type. Array's use [] square bracket notation with elements separated by a comma.
- Object aka dictionary : unordered associative collection. Use {} curly bracket notation with pairs separated by a comma. Within each pair separation is established with a : colon. All keys must be strings and must be unique within that object.
- null : empty value.

JSON Example

- An example of a JSON response From StackOverflows API
- The root object is an object/ dictionary denoted by the curly bracket
- The first pairing is the key “items” with a corresponding array with just one item, a dictionary.
- For this api, each question is represented by a dictionary in this array. I only asked for one.


```
{
  "items": [
    {
      "tags": [
        "ios",
        "objective-c",
        "uiwebview",
        "uiscrollview",
        "screen-orientation"
      ],
      "owner": {
        "reputation": 22,
        "user_id": 3751662,
        "user_type": "registered",
        "accept_rate": 40,
        "profile_image": "https://www.gravatar.com/avatar/fb058f5e726691830e2aca67d0201a55?
        "display_name": "Daljeet",
        "link": "http://stackoverflow.com/users/3751662/daljeet"
      },
      "is_answered": false,
      "view_count": 8,
      "answer_count": 1,
      "score": 0,
      "last_activity_date": 1406309661,
      "creation_date": 1406308178,
      "question_id": 24960983,
      "link": "http://stackoverflow.com/questions/24960983/uiwebview-content-is-lost-partia
      "title": "UIWebView content is lost partially on changing orientation?"
    }
  ],
  "has_more": true,
  "quota_max": 10000,
  "quota_remaining": 9995
```

JSON Parsing – interpreting JSON in your code

- You may eventually use third party frameworks to simplify your JSON parsing, but first you need to understand how to write your own parsing.
- It's conceptually similar to parsing through a plist.
- Use the NSJSONSerialization class to convert JSON to Foundation objects and vice versa.

NSJSONSerialization

`+ JSONObjectWithData:options:error:`

Returns a Foundation object from given JSON data.

Declaration

SWIFT

```swift
class func JSONObjectWithData(_ data: NSData!,
                              options opt: NSJSONReadingOptions,
                                error error: NSErrorPointer) -> AnyObject!
```

OBJECTIVE-C

```objc
+ (id)JSONObjectWithData:(NSData *)data
                 options:(NSJSONReadingOptions)opt
                   error:(NSError **)error
```

Parameters

data	A data object containing JSON data.
opt	Options for reading the JSON data and creating the Foundation objects. For possible values, see NSJSONReadingOptions.
error	If an error occurs, upon return contains an `NSError` object that describes the problem.

NSJSONSerializationOptions

- NSJONReadingMutableContainers : specifies that arrays and dictionaries are created as mutable objects
- NSJSONReadingMutableLeaves : Specifies that leaf strings in the JSON Object graph are created as instances of NSMutableString
- NSJSONReadingAllowFragments : Specifies that the parses should allow top level objects that are not instance of NSArray or NSDictionary
- NSJSONWritingPrettyPrinted: Specifies that the JSON data generated should use white space to make it more human readable.

JSON Parsing & Safety

- We always want to bake a lot of safety into our JSON handling because sometimes the internet/servers don't work as we expect them to work.
- We need to check if the root object is the type are we excepting for that specific API call, and then check to make sure the objects we pull out of it are the types we except too.
- So if the API endpoint we are hitting usually returns a root level dictionary but it returns an array, thats an indicator that something went wrong.
- First checking the HTTP Status Code greatly reduces the amount of errors our JSON parsing will run into, but its better to be safe than sorry.
- Hooray Optionals!

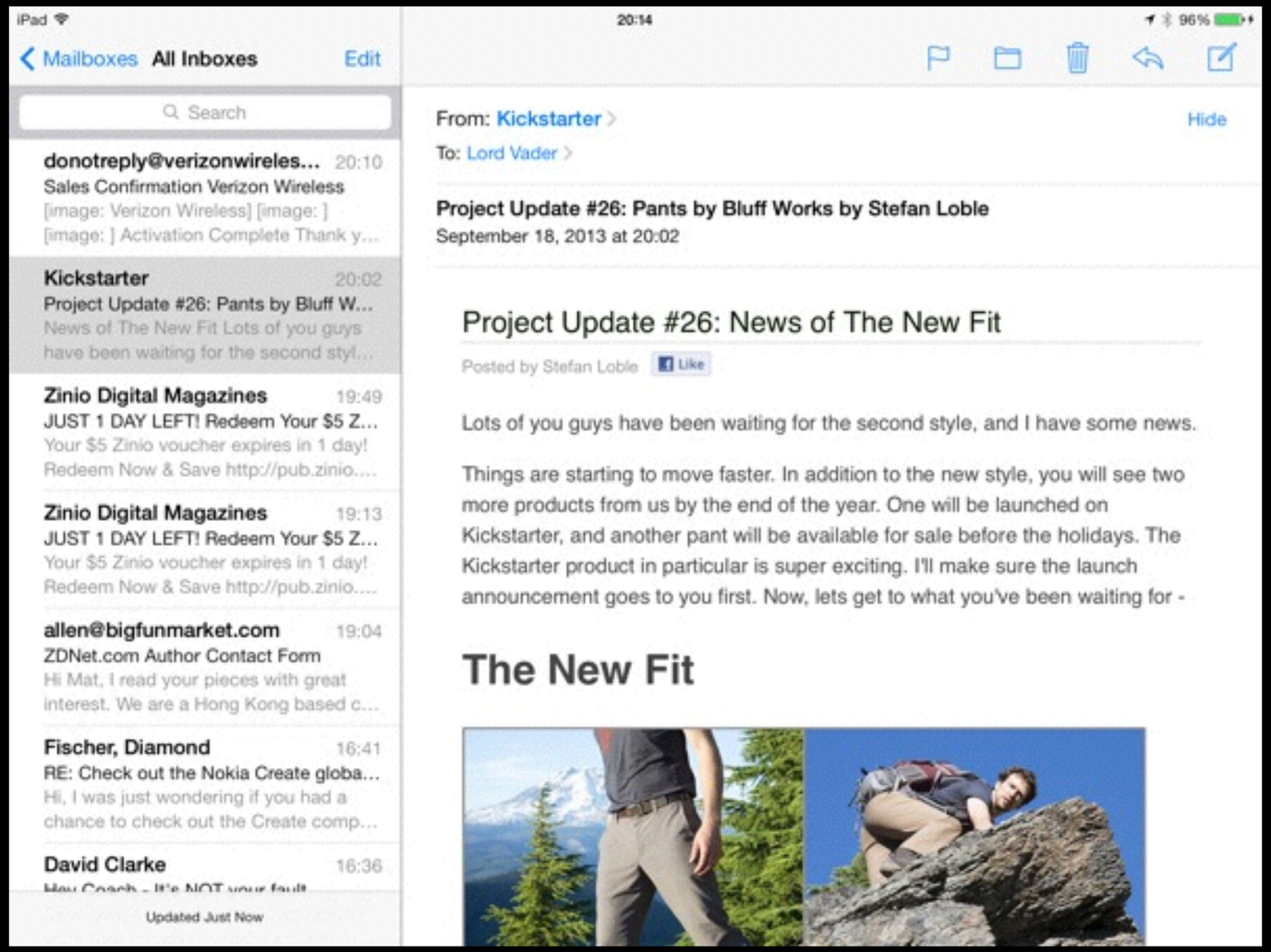
JSON and Optionals

- You are going to use *Optional Binding* a lot when parsing through JSON.
- Optional Binding can be used to find out if an optional contains a value, and if it does then it makes that value available as a temporary constant or variable.
- Example:
- if let items = responseJSON[“items] as? NSArray
- Here we are using Optional Binding combined with type casting to essentially say “If there is an object for key “items” in the responseJSON dictionary, AND its of type NSArray, create a variable called items for this object.

Adaptive Design

- With iOS8, Apple is making a push to simplify the creation of Universal apps.
- All of their standard view controllers are now designed to be used by both devices with the same code.
- An example of this is the UISplitViewController, prior to iOS8 it was only available for iPad. Now it is universal.

UISplitViewController



UISplitViewController

- “UISplitViewController is a container view controller that manages the presentation of two side-by-side view controllers.”
- Typically the left master side will display a list, and the right side presents details of the selected item.
- the UISplitViewController object does not have a user facing interface in itself, it just manages the other view controllers.
- changing the primary and detail view controllers is as simple as calling the appropriate methods:
  - showViewController:sender: or showDetailViewController:sender:
- Some of its functionality is achieved with a delegate

Dynamic Cell Size

- Getting dynamically sizing tableview cells is very easy in iOS8
- Before you had to manually do calculations based on how much text you were getting and then set the height individually for each cell.
- Now all you need to do is properly setup your cells using `autolayout`, and then `tableView.rowHeight` to UITableViewAutomaticDimension.

UISplitViewController

- “UISplitViewController is a container view controller that manages the presentation of two side-by-side view controllers.”
- Typically the left master side will display a list, and the right side presents details of the selected item.
- the UISplitViewController object does not have a user facing interface in itself, it just manages the other view controllers.
- changing the primary and detail view controllers is as simple as calling the appropriate methods:
  - showViewController:sender: or showDetailViewController:sender:
- Some of its functionality is achieved with a delegate

UISearchBar

- Very similar to UITextField
- Delegate methods for when the search and cancel buttons are clicked
- also delegate methods for whenever the text in the search bar is changed if you want to be validating it.
- Same didBeginEditing and didEndEditing methods like the textfield
- Can be embedded inside a tableview for easy interface layout