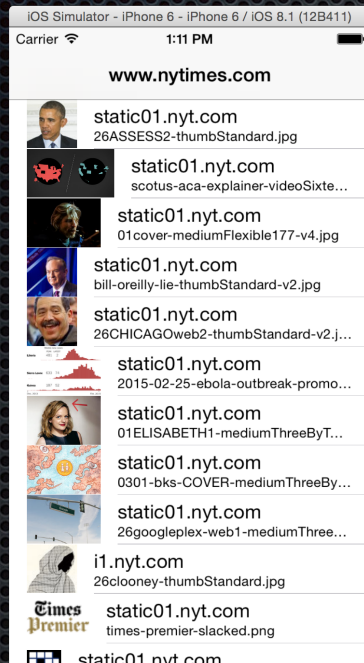
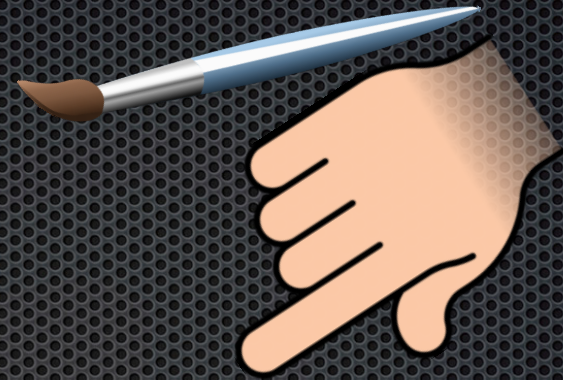


Mobile Application Programming

Asynchronous Communication

Asynchrony

UI



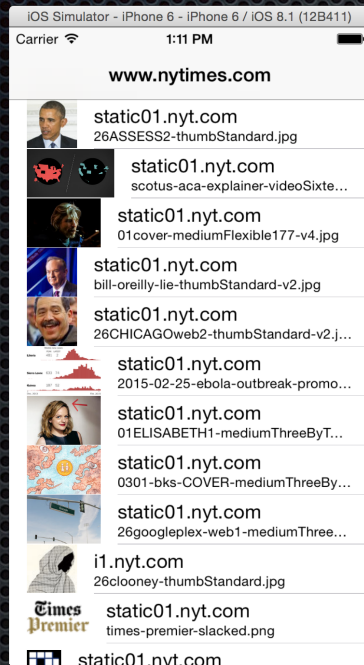
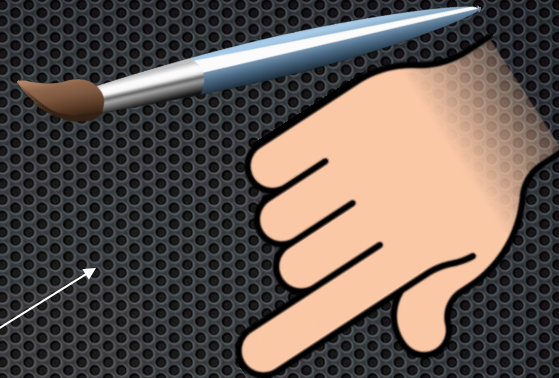
App

Server



Asynchrony

UI



App

Drawing & Touch

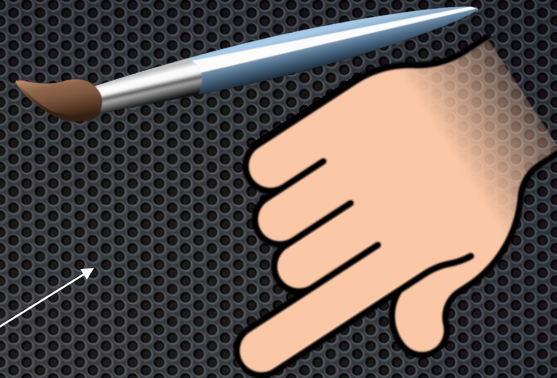
Data
Upload / Download

Server



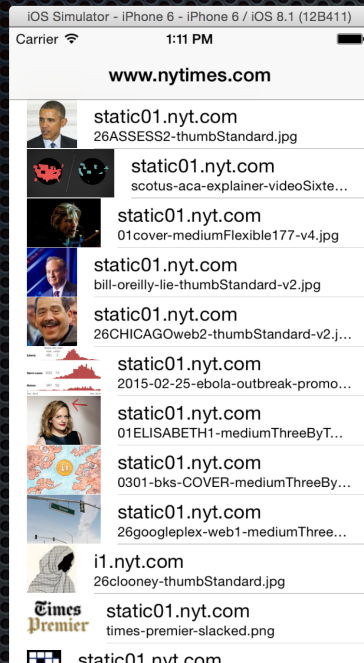
Asynchrony

UI



UI Thread

Drawing & Touch



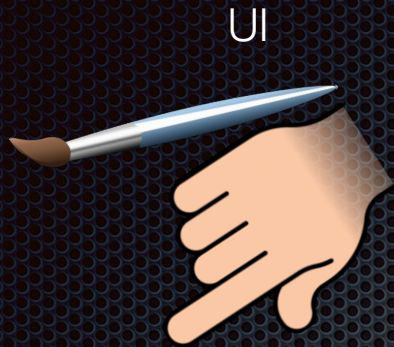
App

Web Thread

Data
Upload / Download

Server





Running ImageCrawler on iPhone 6

ImageCrawler > ImageCrawler > ImageListViewController.swift > viewDidLoad()

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)

0 ImageCrawler.ImageListViewCo...
1 @objc ImageCrawler.ImageListV...
2 -[UITableView _createPreparedC...
3 -[UITableView _updateVisibleCel...
4 -[UITableView layoutSubviews]
5 -[UIView(CALayerDelegate) layo...
6 -[CALayer layoutSublayers]
7 CA::Layer::layout_if_needed(CA...
8 CA::Layer::layout_and_display_i...
9 CA::Context::commit_transactio...
10 CA::Transaction::commit()
11 _afterCACommitHandler
12 __CFRUNLOOP_IS_CALLING_...
13 __CFRunLoopDoObservers
14 __CFRunLoopRun
15 CFRunLoopRunSpecific
16 GSEventRunModal
17 UIApplicationMain
18 top_level_code

```
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64
```

```
imageTableView.dataSource = self  
}  
  
override func didReceiveMemoryWarning()  
{  
    super.didReceiveMemoryWarning()  
    // Dispose of any resources that can be recreated.  
}  
  
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int  
{  
    return _crawler.imageCount  
}  
  
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell  
{  
    let imageIndex: Int = indexPath.row  
    let imageUrl: NSURL = _crawler.imageUrlAtIndex(imageIndex)  
    let imageData: NSData = _crawler.imageAtIndex(imageIndex)  
    let image: UIImage? = UIImage(data: imageData)  
56  
57  
58  
59  
60  
61  
62  
63  
64
```

let cell: UITableViewCell = UITableViewCell(style: UITableViewCellStyle.Subtitle, reuseIdentifier: nil)
cell.imageView?.image = image
cell.textLabel?.text = imageUrl.host
cell.detailTextLabel?.text = imageUrl.lastPathComponent
return cell
}

Thread 1: breakpoint 1.1

0 ImageCrawler.ImageListViewController...ndexPath) -> ObjectiveC.UITableViewCell

Web Request



Running ImageCrawler on iPhone 6

ImageCrawler > ImageCrawler > ImageCrawler.swift > No Selection

ImageCrawler
PID 6106, Paused

CPU 10%

Memory 24.9 MB

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: src (serial)

Thread 2
Queue: com.apple.libdispatch-mana...

Thread 3
Queue: com.apple.root.default-qos (...

Thread 4
Queue: com.apple.root.default-qos (...

Thread 5
Queue: com.apple.NSURLSession-w...

com.apple.NSURLConnectionL...

Thread 6
Queue: NSOperationQueue 0x7fa113...

0 ImageCrawler.ImageCrawler(pa...
1 reabstraction thunk helper from...
2 _67+[NSURLConnection send...

```
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89
```

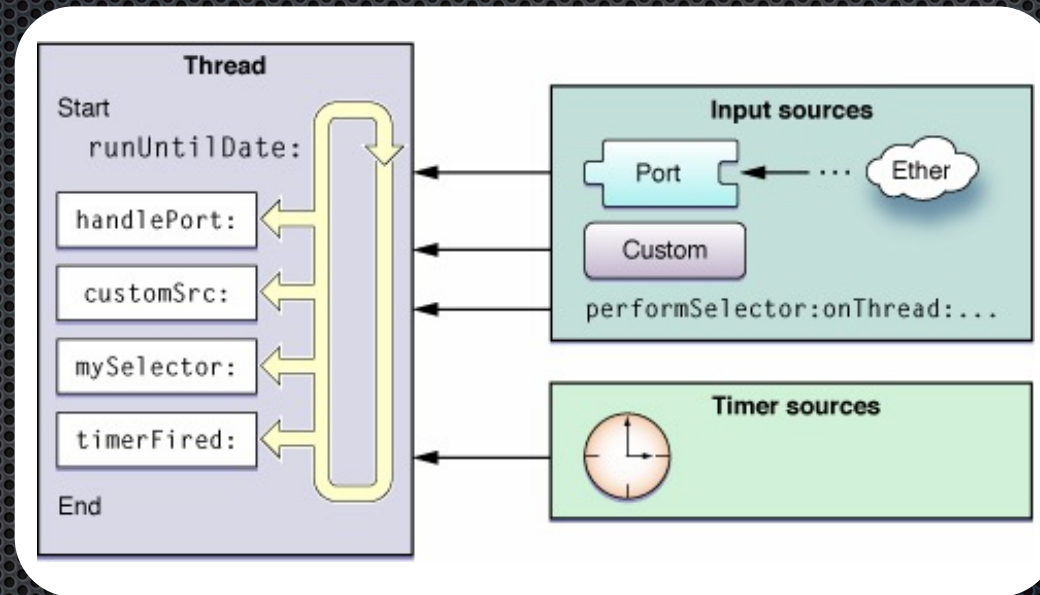
```
let urlString: String = content.substringToIndex(imageTagEnd.location)  
let imageUrl: NSURL? = NSURL(string: urlString)  
if (imageUrl != nil)  
{  
    println("Found Image URL: \(imageUrl!)")  
    imageUrls.append(imageUrl!)  
}  
}  
  
let imageDownloadQueue: NSOperationQueue = NSOperationQueue()  
for imageUrl in imageUrls  
{  
    let imageRequest: NSURLRequest = NSURLRequest(URL: imageUrl)  
    NSURLConnection.sendAsynchronousRequest(imageRequest, queue: imageDownloadQueue)  
    { [weak self] (response: NSURLResponse!, imageData: NSData!, error: NSError!) in  
        if (imageData != nil && imageData!.length > 1024)  
        {  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89
```

NSOperationQueue.mainQueue().addOperationWithBlock()
{
 println("Downloaded Image! Size: \(imageData!.length)")
 self?._imageUrls.append(imageUrl)
 self?._imageCache.updateValue(imageData!, forKey: imageUrl)
 self?._foundImageClosure?()
}
}
}
}
}

Thread 6: breakpoint 1.1

0 ImageCrawler.ImageCrawler.(parseUrl (...-> (ObjectiveC.NSURL -> ()).(closure #1)

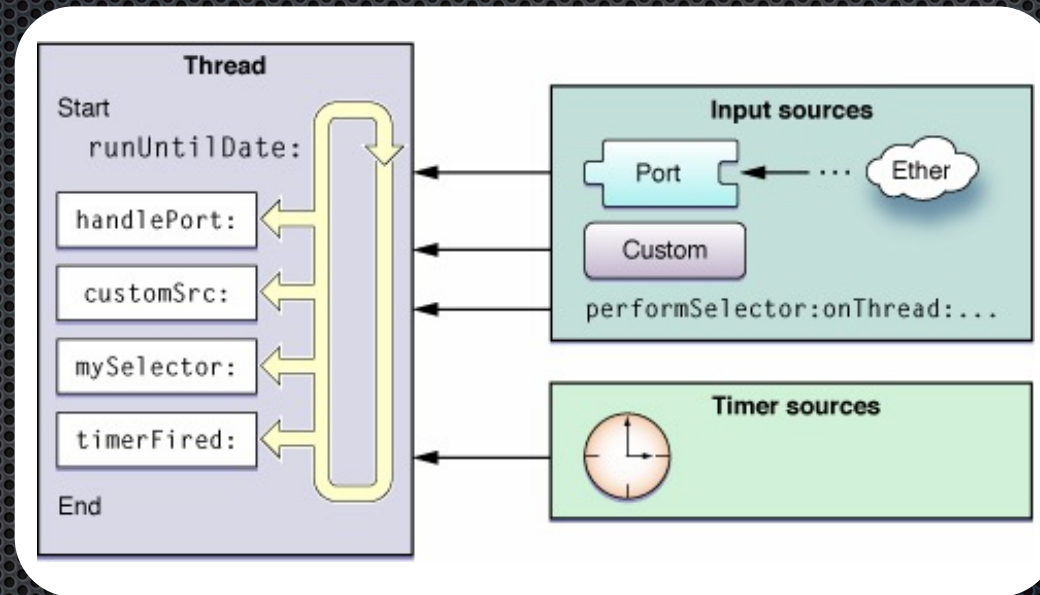
Threads and Run Loops



NSThread / NSRunLoop

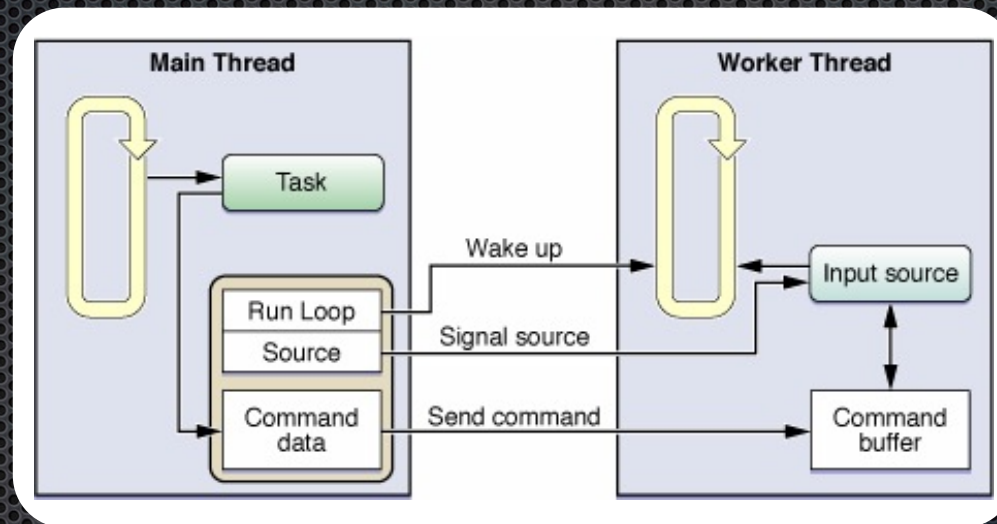
Also see [NSOperationQueue](#) and
[Grand Central Dispatch](#)

Threads and Run Loops



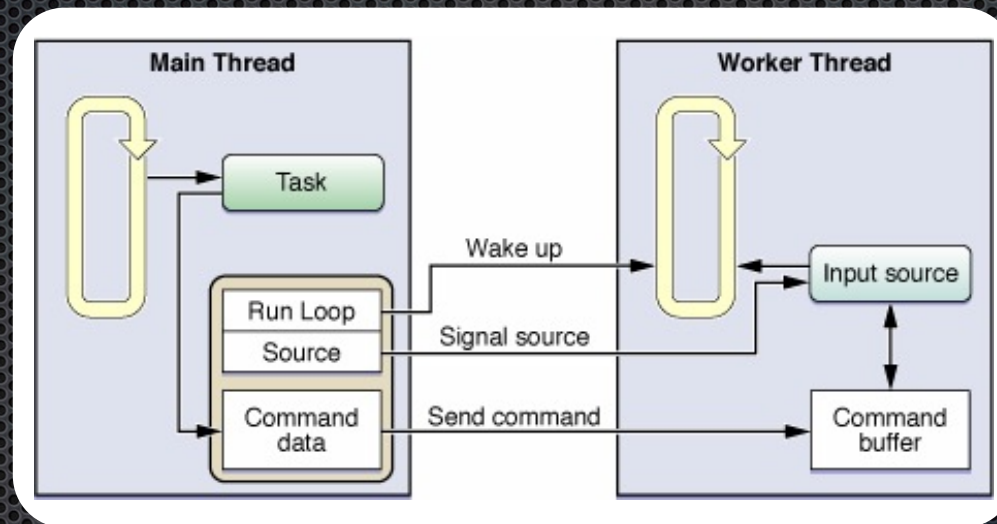
Can only do one task at a time!

Threads and Run Loops

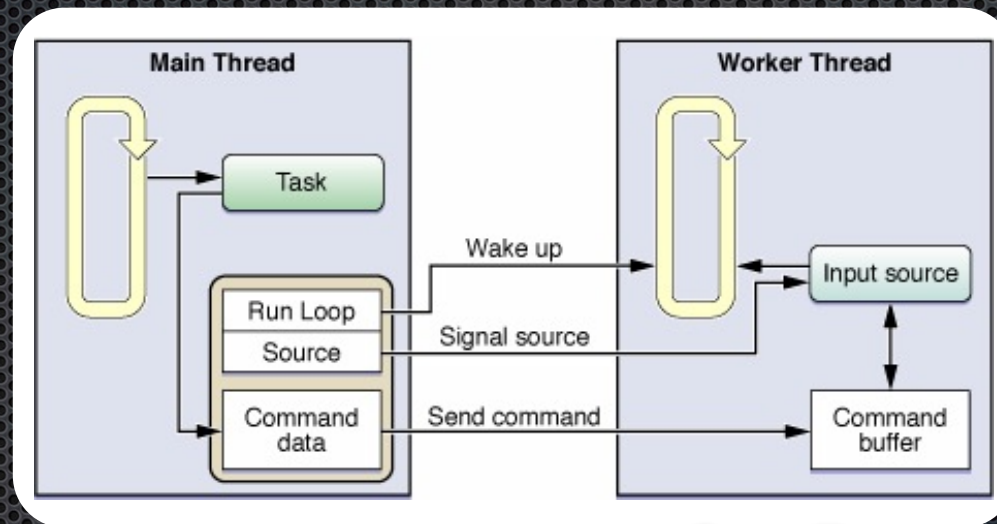


Threads and Run Loops

Get Data at URL

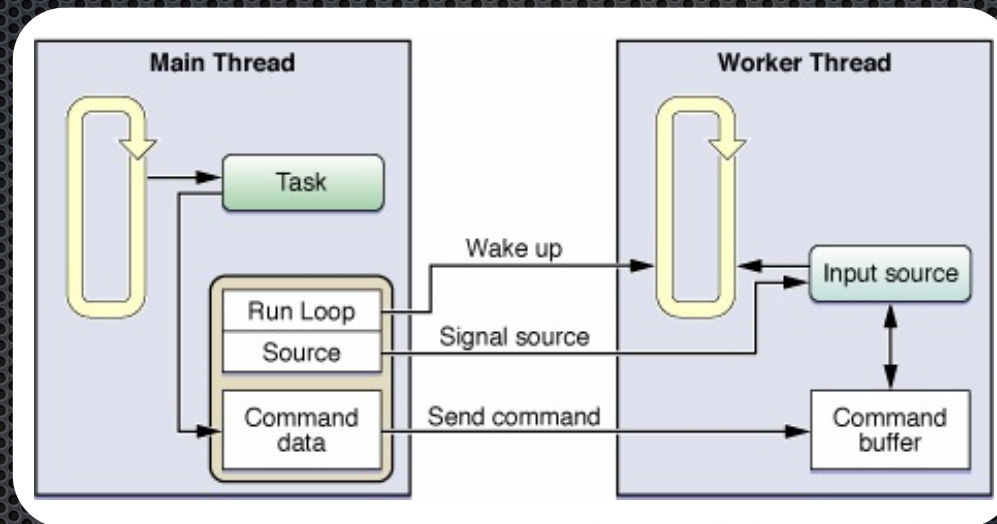


Threads and Run Loops



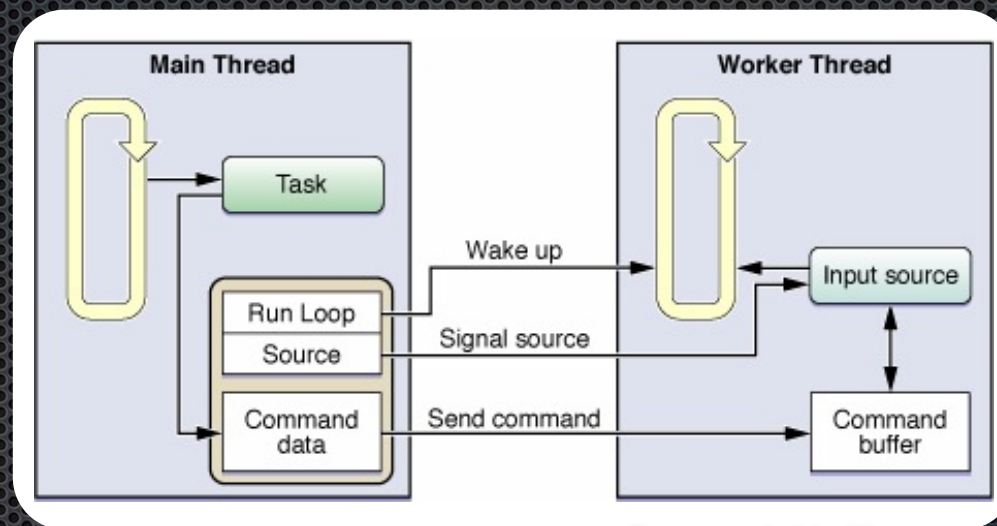
Got Data

Threads and Run Loops



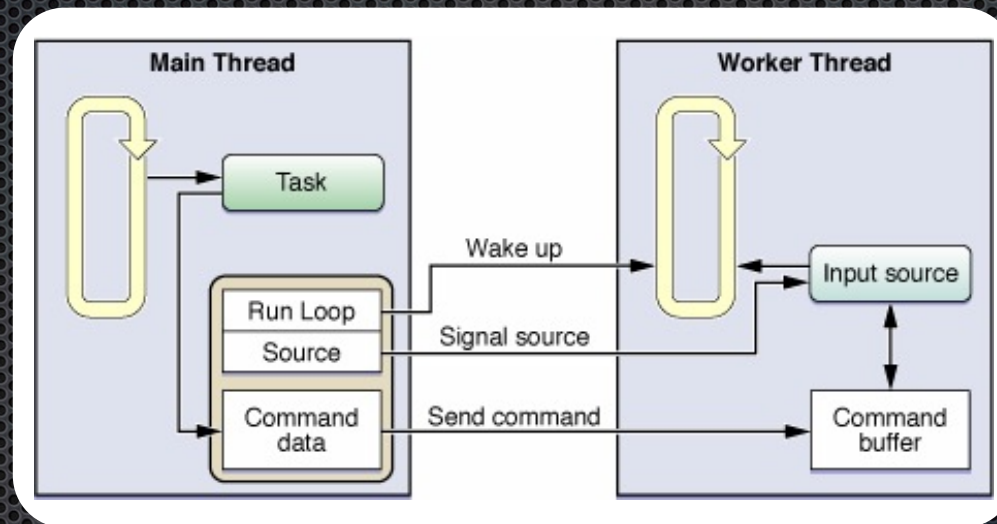
Got More Data

Threads and Run Loops



Got All Data

Threads and Run Loops



Update UI

HTTP



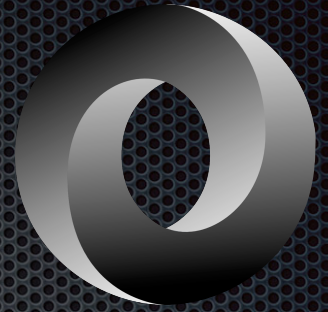
- ✦ Retrieving information from the internet using HTTP
 - ✦ `NSURLRequest` - Collects data about a request
 - ✦ `NSURLConnection` - Executes a request
 - ✦ `NSHTTPURLResponse` - Contains response codes & header
 - ✦ `NSData` - Contains the actual data received

HTTP - Asynchronous



- ✦ Retrieving content **takes time!** Need to thread work.
- ✦ **NSURLConnection** - Has facilitates for threading work
 - ✦ Uses **Grand Central Dispatch** internally
 - ✦ Offers delegate with methods to handle task **setup**, **execution**, **progress updates**, and **finish actions**

Cross-Platform? JSON



Read

```
let d: NSDictionary = NSJSONSerialization.JSONObjectWithData(jsonData, options: .allZeros, error: nil)
let c: ConcreteObject = ConcreteObject(dictionary: d)
```

Write

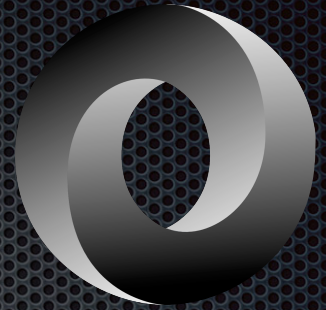
```
let d: NSDictionary = c.dictionaryRepresentation
let jsonData: NSData = NSJSONSerialization.dataWithJSONObject(d, options: .allZeros, error: nil)
```

Reads / Writes

Property List objects **except** **NSData** and **NSDate**

(Resultant data object still needs to be written to file or network connection)

Cross-Platform? JSON



An object that may be converted to JSON must have the following properties:

- The top level object is an `NSArray` or `NSDictionary`.
- All objects are instances of `NSString`, `NSNumber`, `NSArray`, `NSDictionary`, or `NSNull`.
- All dictionary keys are instances of `NSString`.
- Numbers are not NaN or infinity.

Other rules may apply. Calling `isValidJSONObject:` or attempting a conversion are the definitive ways to tell if a given object can be converted to JSON data.