Mobile Application Programming

Server Interaction
Server Interaction

- **Request** a resource using HTTP and a URL
- **Wait** for response data to come back from the server
- **Scrub** the data
  - Remove superfluous parts
  - Validate the data
- **Present** the data to the user
Request and Wait

**NSString (NSData, NSArray, NSDictionary)**

```swift
var s: NSString = NSString(contentsOfURL: url, encoding: NSUTF8StringEncoding, error: nil)
```

**NSURLConnection**

```swift
var data: NSData = URLConnection.sendSynchronousRequest(request, returningResponse: nil, error: nil)
```
Scrub

Flat Data
Scrub

NSString

func `rangeOfString`: NSRange
func `rangeOfCharacterFromSet`: NSRange
func `substringToIndex`: NSString
func `substringFromIndex`: NSString

Also see Regular Expressions and NSPredicate
Present
Cleaner Way?
let parser = NSXMLParser(contentsOfURL: NSURL(string: "http://www.someplace.com"))
parser?.delegate = self
parser?.parse()

func parserDidStartDocument(parser: NSXMLParser!) { /* ... */ }
func parserDidEndDocument(parser: NSXMLParser!) { /* ... */ }
func parser(parser: NSXMLParser!, didStartElement elementName: String!, namespaceURI: String!, qualifiedName qName: String!, attributes attributeDict: [NSObject : AnyObject]!) { /* ... */ }
func parser(parser: NSXMLParser!, foundCharacters string: String!) { /* ... */ }
func parser(parser: NSXMLParser!, didEndElement elementName: String!, namespaceURI: String!, qualifiedName qName: String!) { /* ... */ }
Next Time: REST

- Representational State Transfer (REST)
- Software architectural style used in protocols like HTTP

<table>
<thead>
<tr>
<th>Resource</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection URI, such as <a href="http://example.com/resources/">http://example.com/resources/</a></td>
<td>List the members of the collection, complete with their member URIs for further navigation. For example, list all the cars for sale.</td>
<td>Meaning defined as &quot;replace the entire collection with another collection&quot;.</td>
<td>Create a new entry in the collection where the ID is assigned automatically by the collection. The ID created is usually included as part of the data returned by this operation.</td>
<td>Meaning defined as &quot;delete the entire collection&quot;.</td>
</tr>
<tr>
<td>Element URI, such as <a href="http://example.com/resources/7HOU57Y">http://example.com/resources/7HOU57Y</a></td>
<td>Retrieve a representation of the addressed member of the collection expressed in an appropriate MIME type</td>
<td>Update the addressed member of the collection or create it with the specified ID.</td>
<td>Treats the addressed member as a collection in its own right and creates a new subordinate of it.</td>
<td>Delete the addressed member of the collection.</td>
</tr>
</tbody>
</table>

See chapter 5 of Roy T. Fielding’s doctoral dissertation and REST article on Wikipedia