

# iOS App Development

## 3.1 Accessing Web Resources



# Agenda

- iOS Networking
- Common Web Response Formats
- NSXMLParser
- NSJSONSerialization



# iOS Networking APIs

- **NSURL ( based on CFURL )**
- CFHTTPStream, CFFTPStream
- CFNetwork Framework
- Sockets
- Bonjour: NSNetServices/  
CFNetServices



# Common Web Response Formats

- **REST**
- XML-RPC
- SOAP
- **JSON**
- PHP



# Parsing Web Response

- NSXMLParser class
  - Generic XML Parser
  - Comprehensive control
- **NSJSONSerialization**
  - JSON <-> Foundation Object



# Parsing Web Response

- Key Path (Key-Value Coding)
  - valueForKeyPath:@"places.place"
  - To-many property access returns NSArray
- Downloading text resource at URL
  - [NSString  
+stringWithContentsOfURL:encoding:error:]



# Parsing Web Response

- REST Example Response

```
<?xml version="1.0" encoding="utf-8" ?>  
<rsp stat="ok">  
  <method>flickr.test.echo</method>  
  <format>rest</format>  
  <foo>bar</foo>  
  <api_key>1dcbfa0cf0c4afbbd810e8ce51d3df15</api_key>  
</rsp>
```



# NSXMLParser

- Sample Code from Apple
  - “LazyTableImages”
    - NSXMLParser
    - NSURL Download
    - NSOperation



# Parsing Web Response

- JSON Example Response

```
jsonFlickrApi({
  "stat": "ok",
  "blogs": {
    "blog": [
      {
        "id"      : "73",
        "name"     : "Bloxus test",
        "needspassword" : "0",
        "url"      : "http://remote.bloxus.com/"
      },
      {
        "id"      : "74",
        "name"     : "Manila Test",
        "needspassword" : "1",
        "url"      : "http://flickrtest1.userland.com/"
      }
    ]
  }
})
```



# NSJSONSerialization class

- JSON <-> Foundation Object
- Top Level Object: NSArray or NSDictionary
- All objects are NSString, NSNumber, NSArray, NSDictionary, or NSNull
- Dictionary Keys are NSString
- Numbers are not NaN or infinity



# NSJSONSerialization class

- JSON Data -> Object

```
+ (NSDictionary *) JSONObjectFromURL:(NSURL *) url
{
    // Load JSON data from the URL
    NSData *jsonData = [[NSString stringWithContentsOfURL:url
                                                            encoding:NSUTF8StringEncoding
                                                            error:nil]
                        dataUsingEncoding:NSUTF8StringEncoding];

    // Convert JSON data to Foundation object
    NSDictionary *results = nil;
    if ( jsonData ) {
        NSError *error = nil;
        results = [NSJSONSerialization JSONObjectWithData:jsonData
                                                       options:0
                                                       error:&error];
    }
    return results;
}
```

- + isValidJSONObject:



# Summary

- NSURL/NSString and NSJSONSerialization can be used to access this kind of Web Service API



# References

- URL Loading System Programming Guide
- Key-Value Programming Guide