

iOS



Web Services

Sisoft Technologies Pvt Ltd

SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad

Website: www.sisoft.in Email: info@sisoft.in

Phone: +91-9999-283-283



About Web Services

- Fetch content via HTTP
- Parse the content
- Content is usually
 - XML
 - JSON
- Example: weather data, bus traffic information
Twitter...



Fetching content over HTTP

- In iOS it's easy to fetch content over HTTP
- Classes needed
 - NSURL
 - NSURLRequest
 - NSURLConnection
 - NSURLSession



NSURL

- NSURL object represents a URL that contains the location of resource(file)

```
NSURL *baseURL = [NSURL URLWithString:@"http://www.sisoft.in"];
```

```
NSURL *url = [NSURL URLWithString:@"folder/file.html" relativeToURL:baseURL];
```



NSURLRequest

- Represents a URL Load request
- HTTPMethod may be set. The default HTTP method is “GET”
- The default cache policy is `NSURLRequestUseProtocolCachePolicy`
- The default timeout interval is 60 seconds
- Simply append your key/value pair to the post string



NSURLRequest

- Represents a URL Load request
- HTTPMethod may be set. The default HTTP method is “GET”.
- The default cache policy is `NSURLRequestUseProtocolCachePolicy` and the default timeout interval is 60 seconds
- Simply append your key/value pair to the post string



NSURLRequest

- `NSURL *aUrl = [NSURL URLWithString:@"http://www.apple.com/"];`
- `NSMutableURLRequest *request = [NSMutableURLRequest
requestWithURL:aUrl
cachePolicy:NSURLRequestUseProtocolCachePolicy
timeoutInterval:60.0];`
- `[request setHTTPMethod:@"POST"];`
- `NSString *postString = @"company=Locassa&quality=AWESOME!";`
- `[request setHTTPBody:[postString
dataUsingEncoding:NSUTF8StringEncoding]];`
- `NSURLConnection *connection= [[NSURLConnection alloc]
initWithRequest:request delegate:self];`



NSURLConnection

- Loads the contents of a URL by providing a URL request object
- Provides convenience class methods to load URL requests both asynchronously using a callback block and synchronously
- NSURLConnection Protocols:
 - [NSURLConnectionDelegate](#): credential handling
 - [NSURLConnectionDataDelegate](#): progress information
 - [NSURLConnectionDownloadDelegate](#): Only for downloads



NSURLConnection Example

```
NSURL *url = [NSURL URLWithString: @"http://www.something.com"];
NSURLRequest *req = [NSURLRequest requestWithURL:url];
NSURLConnection connection = [[NSURLConnection alloc] initWithRequest: req
                             delegate:self
                             startImmediately:YES];
```

*connection

A blue arrow originates from the text '*connection' and points upwards and to the right, ending at the 'connection' variable in the third line of the code block above.

NSURLConnectionDelegate



Tasks

Connection Authentication

- `connection:willSendRequestForAuthenticationChallenge:`
- `connection:canAuthenticateAgainstProtectionSpace:`
- `connection:didCancelAuthenticationChallenge:`
- `connection:didReceiveAuthenticationChallenge:`
- `connectionShouldUseCredentialStorage:`

Connection Completion

- `connection:didFailWithError:`



PARSING XML Options

- SAX
 - Event driven
 - Does not consume lot of memory
 - iOS:NSXMLParser
- DOM
 - Reads entire document to memory
 - You can move in the object tree
 - Libxml2-C based API for SAX and DOM



Parsing XML Data using NSXMLParser

- iOS has NSXMLParser for parsing data
- The NSXMLParser opens xml data and calls methods like
 - didStartElement
 - foundCharacters
 - didEndElement



Parsing XML Data using NSXMLParser....

```
NSXMLParser* parser = [[NSXMLParser alloc] initWithData:xmlData];  
[parser setDelegate: self];  
[parser parse];
```

The NSXMLParserDelegate provides you methods
For handling parsing

Handling XML

- `parserDidStartDocument:`
- `parserDidEndDocument:`
- `parser:didStartElement:namespaceURI:qualifiedName:attributes:`
- `parser:didEndElement:namespaceURI:qualifiedName:`
- `parser:didStartMappingPrefix:toURI:`
- `parser:didEndMappingPrefix:`
- `parser:resolveExternalEntityName:systemID:`
- `parser:parseErrorOccurred:`
- `parser:validationErrorOccurred:`
- `parser:foundCharacters:`
- `parser:foundIgnorableWhitespace:`
- `parser:foundProcessingInstructionWithTarget:data:`
- `parser:foundComment:`
- `parser:foundCDATA:`

JSON(JavaScript Object Notation)



- JSON is a lightweight data-interchange format
- It is easy for humans to read and write
- It is easy for machines to parse and generate
- JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others
- JSON is built on two structures:
 - A collection of name/value pairs (mostly used)
 - An ordered list of values



NSJSONSerialization

- You use the NSJSONSerialization class to convert JSON to Foundation objects and convert Foundation objects to 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



Parsing JSON from the data

- We can use the method `dataWithJSONObject:options:error:` to parse a json object to data
- Ex- `NSDictionary* Data = [NSJSONSerialization JSONObjectWithData:responseData options:kNilOptions error:&error];`
 - where response data is json data
 - Option is parsing option
 - error for error message



NSJSONReadingOptions

- **kNilOptions:**
 - kNilOptions is just a constant for 0 which means no parsing option is chosen
- **NSJSONReadingMutableContainers:**
 - The arrays and dictionaries created will be mutable. Good if you want to add things to the containers after parsing it.
- **NSJSONReadingAllowFragments:**
 - Parses out the top-level objects that are not arrays or dictionaries.



Generating JSON Data

- Now lets do opposite ,from NSDictionary lets build some JSON data

- Ex-

```
NSData* jsonData = [NSJSONSerialization  
dataWithJSONObject:info  
options:NSJSONWritingPrettyPrinted  
error:&error];
```

-Where info is NSDictionary with information

-Option for crating jason type

-error for error message



NSJSONWritingOptions

- For the options parameter there's only one possible value – `NSJSONWritingPrettyPrinted`
- if you want to see the JSON use `NSJSONWritingPrettyPrinted` as this will format it nicely
- If you want to send the JSON over the Internet to a server use `kNilOptions` as this will generate compact JSON code
- For example on working with json visit link:

<http://www.raywenderlich.com/5492/>