NSJSONSerialization Tutorial

Based on Chapter 23 (pages 743 - 755) from iOS 5 by Tutorials; refer to that chapter for a more detailed explanation of the steps involved in this tutorial.

Getting Started: Use the project in the *KivaJSONDemo_start* folder.

Part 1

Objective: Access a JSON feed from a web server and manage that data in a structured format.

1. Add the following code to the *viewDidLoad* selector of **ViewController.m**.

```
- (void)viewDidLoad
{
     [super viewDidLoad];
     dispatch_async(kBgQueue, ^{
     NSData* data = [NSData dataWithContentsOfURL: kLatestKivaLoansURL];
     [self performSelectorOnMainThread:@selector(fetchedData:)
withObject:data waitUntilDone:YES];
     });
}
```

What this does: This downloads some JSON formatted content from a webserver. The download happens in an asynchronous thread (in the background), and then calls the selector *fetchedData:* on the main thread passing the data object that was downloaded.

2. In the previous step we defined a selector called *fetchedData* that would be called when the data has been downloaded. Add that selector to **ViewController.m** with the following code.

```
- (void)fetchedData:(NSData *)responseData {
     //parse out the json data
     NSError* error;
     NSDictionary* json = [NSJSONSerialization
          JSONObjectWithData:responseData
          options:kNilOptions
          error:&error];
     NSArray* latestLoans = [json objectForKey:@"loans"];
     NSLog(@"loans: %@", latestLoans);
}
```

What this does: This will convert the JSON formatted data we passed in (responseData) to an NSDictionary object. One of the objects in that dictionary is called **loans**, which is an array of loans. Using NSLog, that array is then being displayed in the console.

3. **RUN THE APP.** This will display a list of the most recent loans as downloaded from Kiva. You'll need to display the console within XCode (⇧⌘C)

4. Add the following code to the bottom of the *fetchedData:* selector.

```
// 1) Get the latest loan
NSDictionary* loan = [latestLoans objectAtIndex:0];
// 2) Get the funded amount and loan amount
NSNumber* fundedAmount = [loan objectForKey:@"funded_amount"];
NSNumber* loanAmount = [loan objectForKey:@"loan_amount"];
float outstandingAmount = [loanAmount floatValue] - [fundedAmount
floatValue];
// 3) Set the label appropriately
humanReadble.text = [NSString stringWithFormat:@"Latest loan: %@ from %@
needs another $%.2f to pursue their entrepreneural dream", [loan
objectForKey:@"name"], [(NSDictionary*)[loan objectForKey:@"location"]
objectForKey:@"country"], outstandingAmount];
```

What this does: This takes the first item in the array of loans that we created in the previous step and creates 3 variables called **fundedAmount**, **loanAmount**, and **outstandingAmount**. Those 3 variables are inserted into a string object and displayed in the **humanReadable** label.

5. **RUN THE APP.** This will display information for a a single loan in the UILabel object on screen in the simulator.

Part 2

Objective: Convert an NSDictionary object to JSON formatted data that can be sent to a web server.

1. Add the following code to the bottom of the *fetchedData:* selector from the previous step.

```
//build an info object and convert to json
NSDictionary* info = [NSDictionary dictionaryWithObjectsAndKeys:
 [loan objectForKey:@"name"], @"who",
 [(NSDictionary*)[loan objectForKey:@"location"] objectForKey:@"country"],
@"where",
[NSNumber numberWithFloat: outstandingAmount], @"what",
nil];

//convert object to data
NSData* jsonData = [NSJSONSerialization dataWithJSONObject:info
options:NSJSONWritingPrettyPrinted error:&error];

//print out the data contents
jsonSummary.text = [[NSString alloc] initWithData:jsonData
encoding:NSUTF8StringEncoding];
```

What this does: This manually creates an NSDictionary object with some data from the *loan* object created in the previous step. Using the *dataWithJSONObject:* selector of the *NSJSONSerialization* class, the dictionary object is then converted to a data object that is actually a JSON formatted string. Note that to use the JSON data object as a string, it needs to be converted using the *initWithData:* selector of an NSString object.

2. **RUN THE APP.** This will display JSON formatted information for a a single loan in the UILabel object on screen in the simulator.