URL LOADING SYSTEM IN OBJECTIVE-C

URL loading is useful in accessing URL, i.e., the items from the internet. It is provided with the help of the following classes:

- NSMutableURLRequest
- NSURLConnection
- NSURLCache
- NSURLAuthenticationChallenge
- NSURLCredential
- NSURLProtectionSpace
- NSURLResponse
- NSURLDownload
- NSURLSession

Here is a simple example for url loading. This cannot be run on command line. We need to create Cocoa Application.

This can be done by selecting New in XCode, then Project and select Cocoa Application under the OS X application section of the window that appears.

Complete the sequence of steps by clicking next and you will be asked to provide a project name and you can give it a name.

The appdelegate.h file will be as follows:

```
#import <Cocoa/Cocoa.h>

@interface AppDelegate : NSObject <NSApplicationDelegate>

@property (assign) IBOutlet NSWindow *window;

@end
```

Update the AppDelegate.m file to following:

```
#import "AppDelegate.h"

@interface SampleClass:NSObject<NSURLConnectionDelegate>
{
    NSMutableData *_responseData;
}
- (void)initiateURLConnection;

@end


@implementation SampleClass

- (void)initiateURLConnection
{
    // Create the request.
    NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL
URLWithString:@"http://date.jsontest.com"]];

    // Create url connection and fire request
```

```
    NSURLConnection *conn = [[NSURLConnection alloc] initWithRequest:request
delegate:self];
    [conn start];
}
#pragma mark NSURLConnection Delegate Methods

- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse
*)response {
    // A response has been received, this is where we initialize the instance var you
created
    // so that we can append data to it in the didReceiveData method
    // Furthermore, this method is called each time there is a redirect so
reinitializing it
    // also serves to clear it
    _responseData = [[NSMutableData alloc] init];
}

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data {
    // Append the new data to the instance variable you declared
    [_responseData appendData:data];
}

- (NSCachedURLResponse *)connection:(NSURLConnection *)connection
                  willCacheResponse:(NSCachedURLResponse*)cachedResponse {
    // Return nil to indicate not necessary to store a cached response for this
connection
    return nil;
}

- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    // The request is complete and data has been received
    // You can parse the stuff in your instance variable now
    NSLog(@"%@",[[NSString alloc]initWithData:_responseData
encoding:NSUTF8StringEncoding]);
}

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error {
    // The request has failed for some reason!
    // Check the error var
}

@end

@implementation AppDelegate

- (void)applicationDidFinishLaunching:(NSNotification *)aNotification
{
    SampleClass *sampleClass = [[SampleClass alloc]init];
    [sampleClass initiateURLConnection];
    // Insert code here to initialize your application
}
@end
```

Now when we compile and run the program, we will get the following result.

```
2013-09-29 16:50:31.953 NSURLConnectionSample[1444:303] {
   "time": "11:20:31 AM",
   "milliseconds_since_epoch": 1380453631948,
   "date": "09-29-2013"
}
```

In the above program, we have created a simple URL connection that takes the time in JSON format and displays the time.