iOS Dev Camp #3 Week 4
Foundation Framework advance

Edward Chiang

2014.10.20 - 2014.10.24

Today

Date & Time

- NSCalendar
- NSDate / NSFormatter
- NSError

Notification

- NSNotification

URL Loading

- NSURL

NSDate

NSDate

- Date objects allow you to represent dates and times in a way that can be used for date calculations and conversions.
- NSDate is one of the fundamental Cocoa value objects. A date object represents an invariant point in time. Because a date is a point in time, it implies clock time as well as a day, so there is no way to define a date object to represent a day without a time.
- To understand how Cocoa handles dates, you must consider NSCalendar and NSDateComponents objects as well.
- Date objects are immutable.

NSDate

- If you want some time other than the current time, you can use one of NSDate's initWithTimeInterval... or dateWithTimeInterval... methods.
- To compare dates, you can use the isEqualToDate:, compare:, laterDate:, and earlierDate: methods.
- Calendars are specified by constants in NSLocale. You can get the calendar for the user's preferred locale most easily using the NSCalendar method currentCalendar; you can get the default calendar from any NSLocale object using the key NSLocaleCalendar.
- Use the dateByAddingComponents:toDate:options: method to add components of a date (such as hours or months) to an existing date.

NSNotification

NSNotification

- Foundation supplies a programming architecture for passing around information about the occurrence of events.
- The standard way to pass information between objects is message passing—one object invokes the method of another object. However, message passing requires that the object sending the message know who the receiver is and what messages it responds to. At times, this tight coupling of two objects is undesirable—most notably because it would join together two otherwise independent subsystems. For these cases, a broadcast model is introduced: An object posts a notification, which is dispatched to the appropriate observers through an NSNotificationCenter object, or simply notification center.

NSNotification

- Any number of objects may receive the notification, not just the delegate object. This precludes returning a value.
- An object may receive any message you like from the notification center, not just the predefined delegate methods.
- The object posting the notification does not even have to know the observer exists.

Registering for Notifications

- You register an object to receive a notification by invoking the notification center method addObserver:selector:name:object:, specifying the observer, the message the notification center should send to the observer, the name of the notification it wants to receive, and about which object.

Posting a Notification

- You can create a notification object with notificationWithName:object: or notificationWithName:object:userInfo:. You then post the notification object to a notification center using the postNotification: instance method. NSNotification objects are immutable, so once created, they cannot be modified.

Unregistering an Observer

- Before an object that is observing notifications is deallocated, it must tell the notification center to stop sending it notifications. Otherwise, the next notification gets sent to a nonexistent object and the program crashes.
- Use the more specific removeObserver... methods that specify the notification name and observed object to selectively unregister an object for particular notifications.

NSURL

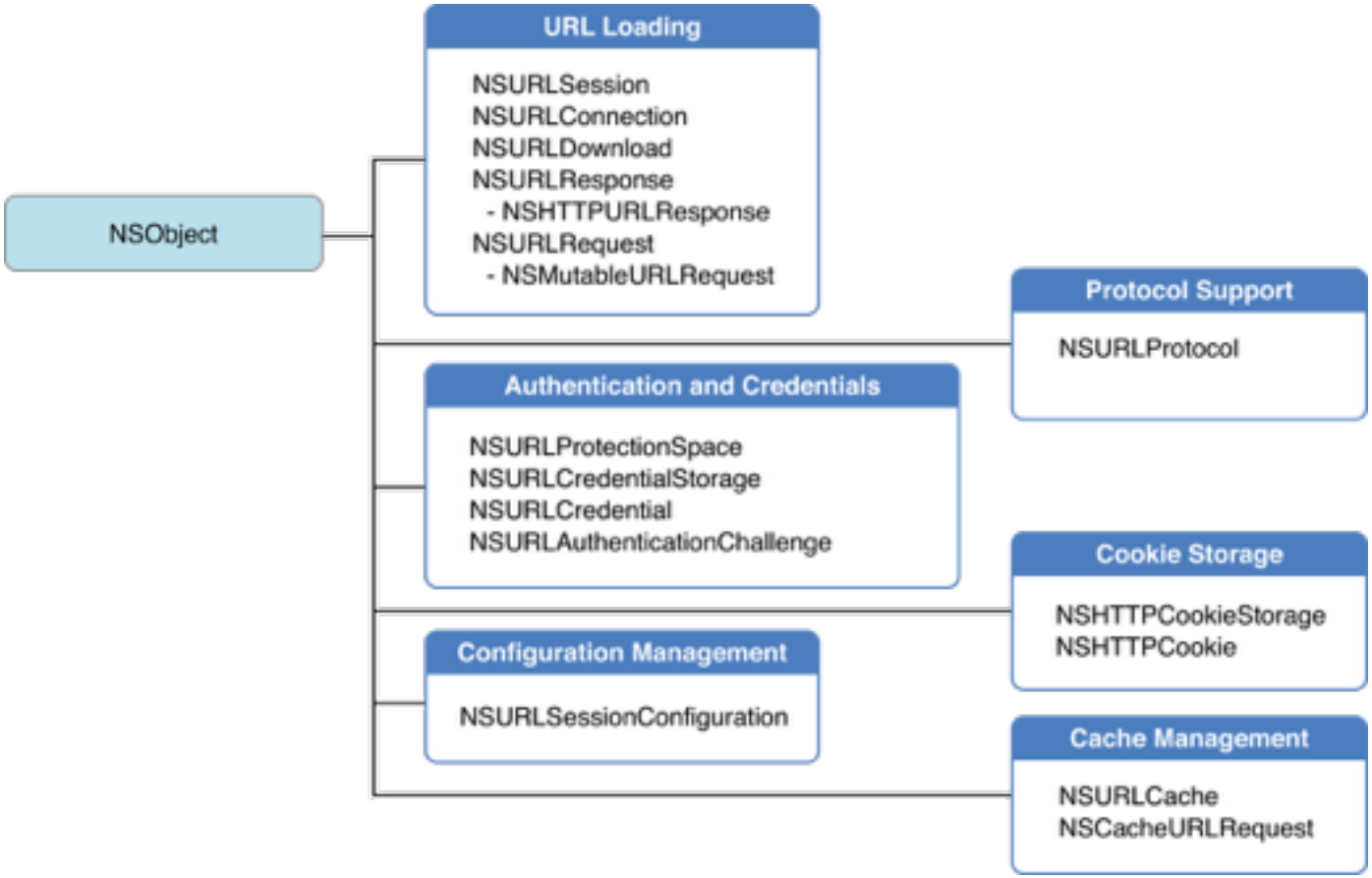
NSURL

- The URL loading system is a set of classes and protocols that allow your app to access content referenced by a URL. At the heart of this technology is the NSURL class, which lets your app manipulate URLs and the resources they refer to.
- To support that class, the Foundation framework provides a rich collection of classes that let you load the contents of a URL, upload data to servers, manage cookie storage, control response caching, handle credential storage and authentication in app-specific ways, and write custom protocol extensions.

NSURL

- The URL loading system provides support for accessing resources using the following protocols:
  - File Transfer Protocol (ftp://)
  - Hypertext Transfer Protocol (http://)
  - Hypertext Transfer Protocol with encryption (https://)
  - Local file URLs (file:///)
  - Data URLs (data://)

NSURL


NSObject
URL Loading
NSURLSession
NSURLConnection
NSURLDownload
NSURLResponse
- NSHTTPURLResponse
NSURLRequest
- NSMutableURLRequest
Protocol Support
NSURLProtocol
Authentication and Credentials
NSURLProtectionSpace
NSURLCredentialStorage
NSURLCredential
NSURLAuthenticationChallenge
Cookie Storage
NSHTTPCookieStorage
NSHTTPCookie
Configuration Management
NSURLSessionConfiguration
Cache Management
NSURLCache
NSCacheURLRequest

The URL loading system

- URL Loading
- Fetching Content as Data (In Memory)
- Downloading Content as a File
- Response Metadata
- Redirection and other request changes
- Authentication and credentials
- Cach Management
- Cookie Storage
- Protocol Support

~ END ~

http://www.alphacamp.tw