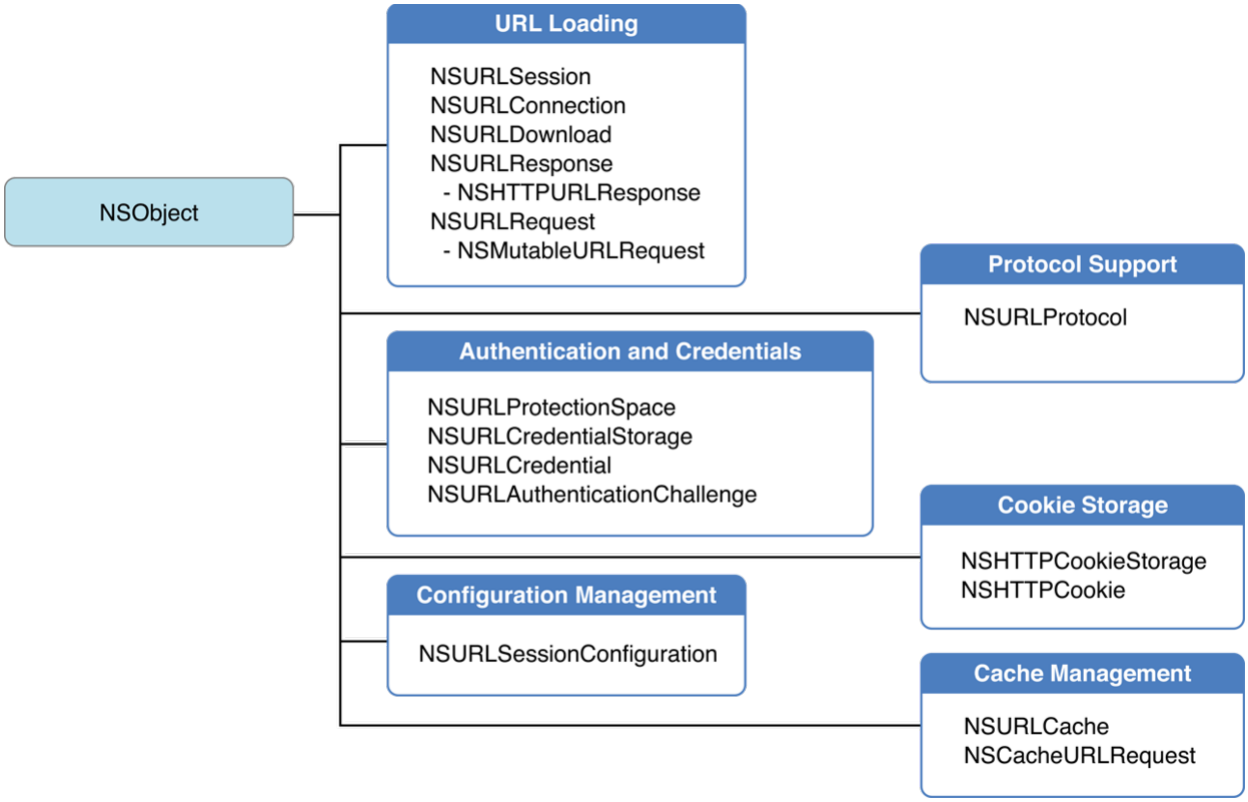
作者 FlyElephant (/users/24da48b2ddb3) 2016.01.24 13:50*
写了31653字，被163人关注，获得了225个喜欢
(/users/24da48b2ddb3)

iOS开发-NSURLSession简介

Core Foundation中NSURLConnection在2003年就随着Safari一同发布，诞生在十多年前，由于iOS设备硬件和软件升级比较快，原有的NSURLConenection网络请求的性能需要重构，2013年iOS7中给出了新的网络请求方案-NSURLSession。日常的iOS开发中，AFNetWorking算是最常见的网络请求库，AFNetworking 1.0建立在NSURLConnection的基础之上,AFNetworking 2.0使用NSURLConnection基础API，以及较新基于NSURLSession的API的选项,3.X版本删除了所有基于NSURLConnection API的所有支持，新的API完全基于NSURLSession。
关于NSURLSession中日常开发中用的比较少，但是我们还是有必要了解一些NSURLSession的基础知识。

基础知识

无论我们使用NSURLConnection还是NSURLSession,最终的目的都是为了获取网络数据和上传数据,单纯的从App使用角度来看获取数据使用的比例在80%以上,用户上传数据使用的比例在20%以下。苹果的网络请求支持URL加载,支持常用http,https，ftp,file,data协议。数据通信用到类和协议可以参考下图:



NSURLSession.png

API详解

本文主要介绍上图中的URL Loading的NSURLSession,NSURLSession需要用到三种异步的任务NSURLSessionDataTask,NSURLSessionDownloadTask和NSURLSessionUploadTask。

NSURLSessionDataTask主要用于获取JSON/XML数据；

NSURLSessionDownloadTask的作用是下载文件;

NSURLSessionUploadTask上传文件会用到;

首先来看一下NSURLSessionDataTask工作模式:

```
NSURL *url=[NSURL URLWithString:@"http://www.jianshu.com/users/24da48b2ddb3/latest_a
NSURLRequest *urlRequest=[NSURLRequest requestWithURL:url];
NSURLSession *urlSession=[NSURLSession sharedSession];
NSURLSessionDataTask *dataTask=[urlSession dataTaskWithRequest:urlRequest completionH
    NSDictionary *content = [NSJSONSerialization JSONObjectWithData:data options:NSJS
    NSLog(@"%@",content);
}];
[dataTask resume];
```

代码中我们发现NSURLSession通过单例模式获取了全局共享的cache,cookie和证书,任务完成调用之后我们只需要在block中进行调用数据验证判断即可。
如果需要上传我们将上面的NSURLSessionDataTask替换成NSURLSessionUploadTask即可:

```
NSURLSessionUploadTask *uploadTask=[urlSession uploadTaskWithRequest:urlRequest fromD

}];
```

委托方式

我们通过在block中进行数据处理，同样的NSURLSession的任务也可以通过设置delegate来完成数据操作:

```
NSURLSessionConfiguration *sessionConfig = [NSURLSessionConfiguration defaultSessionC
NSURLSession *inProcessSession = [NSURLSession sessionWithConfiguration:sessionConfig
NSString *url = @"http://www.jianshu.com/users/24da48b2ddb3/latest_articles";
NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL URLWithString:url]];
NSURLSessionDownloadTask *downloadTask = [inProcessSession downloadTaskWithRequest:re
[downloadTask resume];
```

代码中NSURLSessionConfiguration有三种模式:
1.defaultSessionConfiguration:默认的进程内模式
2.ephemeralSessionConfiguration:短暂的(内存)，进程内模式
3.backgroundSessionConfigurationWithIdentifier:后台模式，在iOS8可以使用,后台接收完成之后可以在APPDelegate进行处理:

```
-(void)application:(UIApplication )application handleEventsForBackgroundURLSession:(N
```

NSURLSession设置委托为self,还可以是设置队列,默认是在主队列中以非阻塞线程的方式更新数据和UI,获取数据的代码我们只需要修改dataTask为download形式就可以进行文件下载。
NSURLSessionDelegate中在完成任务之后调用:

```
-(void)URLSession:(NSURLSession )session task:(NSURLSessionTask )task didCompleteWitl
```

NSURLSessionDownloadDelegate下载的方法调用:

```
-(void)URLSession:(NSURLSession )session downloadTask:(NSURLSessionDownloadTask )dow
```


NSURLSessionConfiguration还可以缓存协议，超时时间，网络请求类型,SSL,TSL协议等一系列属性，有兴趣的可以自行研究一下~
NSURLRequest对应的NSMutableURLRequest可以设置Http请求的请求头,Host,Accept,Accept-Encoding等附加的Http头部请求信息.
Http请求中有两种方式get和post，NSURLRequest是get方法请求，同样的通过NSMutableRequest的setHTTPMethod设置请求方法.

封装请求

关于NSURLSession的如果用的比较多可以进行深层次的封装,本文简单的Get和Post进行简单封装,主要封装三个方式，get封装，get含参数封装,post封装:

```
typedef void (^CompletioBlock)(NSDictionary dict, NSURLResponse response, NSError *e
@interface FENetWork : NSObject
+(void)requesetWithUrl:(NSString *)url completeBlock:(CompletioBlock)block;
+(void)requesetWithUrl:(NSString )url params:(NSDictionary )params completeBlock:(Co
+(void)requesetWithData:(NSString )url bodyData:(NSData )data completeBlock:(Complet
@end
```

关于方法封装中遇到的一个问题就是关于url编码的问题,其中需要注意的这里的允许字符集是URLQueryAllowedCharacterSet,NSCharacterSet包含各种各种的字符集合,根据选择选择正确的字符容器:

```
- (nullable NSString )stringByAddingPercentEncodingWithAllowedCharacters:(NSCharacte
```

实现代码:

```
#import "FENetWork.h"
@implementation FENetWork
+(void)requesetWithUrl:(NSString *)url completeBlock:(CompletioBlock)block{
    NSString *urlEnCode=[url stringByAddingPercentEncodingWithAllowedCharacters:[NSC
    NSURLRequest *urlRequest=[NSURLRequest requestWithURL:[NSURL URLWithString:urlEn
    NSURLSession *urlSession=[NSURLSession sharedSession];
    NSURLSessionDataTask *dataTask=[urlSession dataTaskWithRequest:urlRequest comple
        if (error) {
            block(nil,response,error);
        }else{
            NSDictionary *content = [NSJSONSerialization JSONObjectWithData:data opt
            block(content,response,error);
        }
    }];
    [dataTask resume];
}
+(void)requesetWithUrl:(NSString )url params:(NSDictionary )params completeBlock:(Co
NSMutableString *mutableUrl=[[NSMutableString alloc]initWithString:url];
if ([params allKeys]) {
    [mutableUrl appendString:@"?"];
    for (id key in params) {
        NSString *value=[[params objectForKey:key] stringByAddingPercentEncodingWith
        [mutableUrl appendString:[NSString stringWithFormat:@"%@=%@&",key,value]];
    }
}
[self requesetWithUrl:[mutableUrl substringToIndex:mutableUrl.length-1] completeBloc
}
+(void)requesetWithData:(NSString *)url bodyData:(NSData *)data completeBlock:(Compl
    NSMutableURLRequest *mutableRequest=[NSMutableURLRequest requestWithURL:[NSURL U
    [mutableRequest setHTTPMethod:@"POST"];
    [mutableRequest setHTTPBody:data];
    NSURLSession *urlSession=[NSURLSession sharedSession];
    NSURLSessionDataTask *dataTask=[urlSession dataTaskWithRequest:mutableRequest co
        if (error) {
            block(nil,response,error);
        }else{
            NSDictionary *content = [NSJSONSerialization JSONObjectWithData:data opt
            block(content,response,error);
        }
    }];
    [dataTask resume];
}@end
```

对此封装方法比较简单,适用于学习,行笔匆匆,难免遗漏,如有不当,欢迎讨论~

推荐拓展阅读 (/sign_in)

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

¥ 打赏支持

♡ 喜欢 | 8

分享到微博 分享到微信
更多分享

0条评论 （ 按时间正序 · 按时间倒序 · 按喜欢排序 ）

添加新评论 (/sign_in)

登录后发表评论 (/sign_in)

被以下专题收入，发现更多相似内容：

iOS Developer (/collection/3233d1a249ca)

分享 iOS 开发的知识，解决大家遇到的问题，讨论iOS开发的前沿，欢迎大家投稿！(/collection/3233d1a249ca)

13304篇文章 (/collection/3233d1a249ca) · 26268人关注

+ | 添加关注 (/sign_in)

iOS开发技巧 (/collection/19dbe28002a3)

【简介】 专题内容主要包括OC、swift等涉及到iOS开发进阶的内容。 swift可以关注下我的另一个专题：swift开发...(/collection/19dbe28002a3)

1053篇文章 (/collection/19dbe28002a3) · 20071人关注

+ | 添加关注 (/sign_in)

iOS 开发 (/collection/2ffaa203eb6a)

7792篇文章 (/collection/2ffaa203eb6a) · 6700人关注

+ | 添加关注 (/sign_in)

(/collection/2ffaa203eb6a)