# 面试准备内容

# 预备工作

## cocoaPods的使用

# 网络操作\_1（AFNetworking）

## get方式

## post方式

## 序列化，get请求序列化

## 添加请求头

## json解析中文乱码解决

[manager POST:@"http://apis.baidu.com/tngou/cook/name" parameters:dict success:^(NSURLSessionDataTask \* \_Nonnull task, id \_Nullable responseObject) {

NSData \*jsonData = [NSJSONSerialization dataWithJSONObject:responseObject options:NSJSONWritingPrettyPrinted error:nil];

// NSData转为NSString

NSString \*jsonStr = [[NSString alloc] initWithData:jsonData encoding:NSUTF8StringEncoding];

## 大文件下载与断点下载

iOS开发之网络编程--使用NSURLConnection实现大文件断点续传下载+使用输出流代替文件句柄

goodboy\_heyang goodboy\_heyang 浏览 8 2016-04-12 17:42:07

iOS开发之网络编程--下载

摘要

前言：本篇讲解，在前篇iOS开发之网络编程--使用NSURLConnection实现大文件断点续传下载的基础上，使用输出流代替文件句柄实现大文件断点续传。 　　 在实际开发中，输入输出流用的比较少，但是用起来也是很方便的。iOS开发用到的输入输出流和在Java中的输入输出流是几乎一样的，本质也是一...

前言：本篇讲解，在前篇iOS开发之网络编程--使用NSURLConnection实现大文件断点续传下载的基础上，使用输出流代替文件句柄实现大文件断点续传。

　　 在实际开发中，输入输出流用的比较少，但是用起来也是很方便的。iOS开发用到的输入输出流和在Java中的输入输出流是几乎一样的，本质也是一个意思：将网络返回的数据当做流来处理。

　　 输入输出的理解：输入到哪里？输出到哪里？这个问题不难理解，输入输出是要站着服务器角度来思考的，下面用图来解释：

代码关键词：

　　 1、在接收到响应头的代理方法里创建输出流(根据上面的图，下载自然需要创建输出流NSOutputStream)。

　　 2、在接收数据的代理方法中写(write)数据，注意写入的是data字节(data.bytes)。

　　　3、最后在下载完毕的代理方法里关闭输出流。

用来做代码练习的API接口：

MP4小视频：http://120.25.226.186:32812/resources/videos/minion\_01.mp4

完整的关键代码：

#import "ViewController.h" @interface ViewController () @property (nonatomic ,assign)NSInteger totalSzie; @property (nonatomic ,assign)NSInteger currentSzie; @property (nonatomic, strong) NSString \*fileName; /\*\* 文件的路径\*/ @property (nonatomic ,strong) NSString \*fullPath; /\*\* 请求对象\*/ @property (nonatomic ,strong)NSURLConnection \*connect; /\*\* 输出流\*/ @property (nonatomic ,strong)NSOutputStream \*stream; @property (weak, nonatomic) IBOutlet UIProgressView \*progressView; @end @implementation ViewController #pragma mark ---------------------- #pragma mark Events - (IBAction)downloadBtnClick:(id)sender { // [[NSFileManager defaultManager] removeItemAtPath:self.fullPath error:nil]; [self download]; } - (IBAction)cancelBtnClick:(id)sender { //取消网络请求 [self.connect cancel]; } #pragma mark ---------------------- #pragma mark Methods -(void)download { NSLog(@"------"); //1.确定url NSURL \*url =[NSURL URLWithString:@"http://120.25.226.186:32812/resources/videos/minion\_01.mp4"]; //2.创建请求对象 NSMutableURLRequest \*request = [NSMutableURLRequest requestWithURL:url]; //设置请求头信息,说明只需要请求该资源的一部分数据 /\* bytes=0-1000 表示下载0~1000的数据 bytes=0- 表示从0开始下载直到下载完毕 bytes=100- 表示从0开始下载直到下载完毕 \*/ NSString \*range = [NSString stringWithFormat:@"bytes=%zd-",self.currentSzie]; [request setValue:range forHTTPHeaderField:@"Range"]; NSLog(@"%@",range); //3.发送异步请求 self.connect = [NSURLConnection connectionWithRequest:request delegate:self]; } #pragma mark ---------------------- #pragma mark NSURLConnectionDataDelegate -(void)connection:(NSURLConnection \*)connection didReceiveResponse:(NSURLResponse \*)response { NSLog(@"--didReceiveResponse-"); //判断是否已经下载过了 if (self.currentSzie >0) { return; } //0.获得文件的总大小 //expectedContentLength是本次请求的数据的大小,并不是整个 self.totalSzie = response.expectedContentLength; //1.得到文件的名称 self.fileName = response.suggestedFilename; //2.获得文件的全路径 //caches NSString \*caches = [NSSearchPathForDirectoriesInDomains(NSCachesDirectory, NSUserDomainMask, YES) lastObject]; NSString \*fullPath = [caches stringByAppendingPathComponent:self.fileName]; self.fullPath = fullPath; //3.创建输出流 /\* 第一个参数: 写入数据的地址 第二个参数: 表示要不要追加 断点续传肯定要追加 \*/ NSOutputStream \*stream = [[NSOutputStream alloc]initToFileAtPath:fullPath append:YES]; self.stream = stream; //4.打开数据流 // 如果文件不存在,那么会自动创建一个空的文件 [self.stream open]; } -(void)connection:(NSURLConnection \*)connection didReceiveData:(NSData \*)data { //写数据 /\* 第一个参数:要写的数据 第二个参数:数据的长度 \*/ [self.stream write:data.bytes maxLength:data.length]; //3.累加当前下载的数据大小 self.currentSzie +=data.length; //4.计算文件的下载进度 NSLog(@"%f",1.0 \* self.currentSzie / self.totalSzie); self.progressView.progress = 1.0 \* self.currentSzie / self.totalSzie; } -(void)connectionDidFinishLoading:(NSURLConnection \*)connection { NSLog(@"%@",self.fullPath); //1.关闭输出流 [self.stream close]; //2.清空指针 self.stream = nil; } -(void)connection:(NSURLConnection \*)connection didFailWithError:(NSError \*)error { } @end

另一种方案网址是：<http://www.cnblogs.com/goodboy-heyang/p/5198810.html>

iOS9之后使用的是nsurlsession 处理

## 监听文件下载进度

## 多文件下载

## 文件上传（图片与文字）

## \_\_block \_\_unsafe\_unretained typeof(self) weakSelf = self;

# 网络操作\_2系统自带（软件开发技术笔记）

# 多线程

## NSThread

## NSOperation

## GCD

# block

# 动画

### 系统自带的core animation

### 较好的第三方动画库

# 数据存储

# autolayout多设备适配

# 推送通知

# 单元测试

# oc重要的理论知识

# UI基本组件

# 后台运行

# 常用的设计模式

### 创建型模式

* 简单工厂模式
* 工厂模式
* 抽象工厂模式
* 单例模式
* 建造者模式
* ~~原型模式~~

### 结构型

### 行为型

# Storyboad

# instruments

# svn、git的使用

# 数据结构与算法

# SWIFT

# 定位与地图

# 国际化

# 下拉刷新，上拉加载更多

# runtime

# 技术总结与记录

# bug解决纪录

## 问题1

**2016-04-18 02:50:40.009 Practice\_Network[903:26081] \*\*\* Terminating app due to uncaught exception 'NSUnknownKeyException', reason: '[<ViewController 0x7fb659f3e0f0> setValue:forUndefinedKey:]: this class is not key value coding-compliant for the key continue.'**

**\*\*\* First throw call stack:**

**(**

**0 CoreFoundation 0x000000010e882f65 \_\_exceptionPreprocess + 165**

**1 libobjc.A.dylib 0x000000010e2fadeb objc\_exception\_throw + 48**

**2 CoreFoundation 0x000000010e882ba9 -[NSException raise] + 9**

**3 Foundation 0x000000010dec6c6b -[NSObject(NSKeyValueCoding) setValue:forKey:] + 288**

**4 UIKit 0x000000010edb64ee -[UIViewController setValue:forKey:] + 88**

**5 UIKit 0x000000010efd65a3 -[UIRuntimeOutletConnection connect] + 109**

**6 CoreFoundation 0x000000010e7c3aa0 -[NSArray makeObjectsPerformSelector:] + 224**

**7 UIKit 0x000000010efd4f86 -[UINib instantiateWithOwner:options:] + 1864**

**8 UIKit 0x000000010edbcf99 -[UIViewController \_loadViewFromNibNamed:bundle:] + 381**

**9 UIKit 0x000000010edbd8c5 -[UIViewController loadView] + 178**

**10 UIKit 0x000000010edbdbc8 -[UIViewController loadViewIfRequired] + 139**

**11 UIKit 0x000000010edbe3c9 -[UIViewController view] + 27**

**12 UIKit 0x000000010ec9a66c -[UIWindow addRootViewControllerViewIfPossible] + 61**

**13 UIKit 0x000000010ec9ad55 -[UIWindow \_setHidden:forced:] + 282**

**14 UIKit 0x000000010ecac6a8 -[UIWindow makeKeyAndVisible] + 42**

**15 UIKit 0x000000010ec27f70 -[UIApplication \_callInitializationDelegatesForMainScene:transitionContext:] + 4131**

**16 UIKit 0x000000010ec2e440 -[UIApplication \_runWithMainScene:transitionContext:completion:] + 1755**

**17 UIKit 0x000000010ec2b61b -[UIApplication workspaceDidEndTransaction:] + 188**

**18 FrontBoardServices 0x000000011204b7ac -[FBSSerialQueue \_performNext] + 192**

**19 FrontBoardServices 0x000000011204bb1a -[FBSSerialQueue \_performNextFromRunLoopSource] + 45**

**20 CoreFoundation 0x000000010e7af031 \_\_CFRUNLOOP\_IS\_CALLING\_OUT\_TO\_A\_SOURCE0\_PERFORM\_FUNCTION\_\_ + 17**

**21 CoreFoundation 0x000000010e7a4f5c \_\_CFRunLoopDoSources0 + 556**

**22 CoreFoundation 0x000000010e7a4413 \_\_CFRunLoopRun + 867**

**23 CoreFoundation 0x000000010e7a3e28 CFRunLoopRunSpecific + 488**

**24 UIKit 0x000000010ec2afb8 -[UIApplication \_run] + 402**

**25 UIKit 0x000000010ec2fca1 UIApplicationMain + 171**

**26 Practice\_Network 0x000000010d8c12df main + 111**

**27 libdyld.dylib 0x00000001107dd92d start + 1**

**28 ??? 0x0000000000000001 0x0 + 1**

**)**

**libc++abi.dylib: terminating with uncaught exception of type NSException**

### 解决方案：

在storyboad的xib界面上出现了问题，我重新删除后，重新拖拽就可以了。