

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#) [更多](#) ▼

您还未登录! [登录](#) [注册](#)



90%的开发者选择极光推送
不仅是集成简单、24小时一对一技术支持

我期望的世界

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)

实战iOS7之后台多任务

博客分类:

- [移动开发](#)

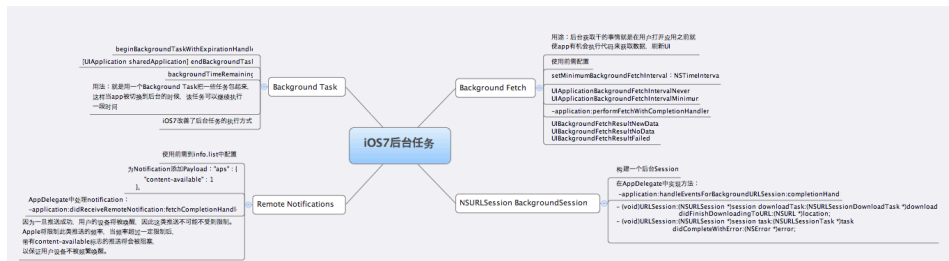
iosObjective-C

在WWDC2013中 水果公司把iOS7的中后台多任务单独开出来一个Session讲述(Session 204 What's New with Multitasking)，其对后台多任务的改动如下:

1. 改变了后台任务的运行方式 (Background Task)
2. 增加了后台获取 (Background Fetch)
3. 增加了推送唤醒 (静默推送, Silent Remote Notifications)
4. 增加了后台传输 (NSURLSession的BackgroundSession)

开发者如果可以更好的利用这些特性，可以很大的提升自己的应用体验，比方说，使用后台获取特性，在你打开微博之前，最新的微博已经自动刷出来，不需要用户再手动刷一次。

知识点



用法

后台任务

后台任务是任务级别的配置，它的作用是当应用切换到后台后仍能保持一段运行时间以完成一些耗时的任务，通常这个时间比较短，不适宜做耗时大的操作，用法

1.启动后台任务

Object-c代码

```
1. [[UIApplication sharedApplication] beginBackgroundTaskWithExpirationHandler:^(
2. ...
3. )];
```

2.结束后台任务

Object-c代码

```
1. [[UIApplication sharedApplication] endBackgroundTask: self.backgroundUpdateTask];
```

后台获取

后台获取是应用级配置，其可以帮助应用在合适的时候触发一次后台获取，我们只能配置最快什么时候做一次获取，但是具体什么时候做，这得看设备的



心情，不过水果公司说，其会根据用户习惯做后台获取，比方说你每天中午12:00使用这个应用，那么它就会在11:50给你做一次后台获取，是不是这么神，有待检验。用法：

- 1.在info.plist中为应用配置后台获取
- 2.在系统启动配置后台获取的（最快）频率
- 3.在AppDelegate中实现方法-application:performFetchWithCompletionHandler: 执行后台获取代码

远程唤醒通知



在iOS7中，对远程通知进行了改造，新的远程通知可以附带一些参数，这样可以减少一些不必要的网络请求，提升性能。它使设备可以接收远端推送后让系统唤醒设备和我们的后台应用，并先执行一段代码来准备数据和UI，然后再提示用户有推送。这时用户如果解锁设备进入应用后将不会再有任何加载过程，新的内容将直接得到呈现。注意，因为，每一次远程通知都会唤醒设备，为了保证设备续航，水果公司对发送通知的频率是有限制的，尽量少用。用法

- 1.为应用配置使用远程推送
- 2.使用最新格式的notification

Json代码  

```
1. {  
2.   "aps": {  
3.     "content-available" : 1  
4.   },  
5.   "content-id" : 42  
6. }
```

- 3.AppDelegate中配置delegate方法处理notification:



Object-c代码  

```
1. -(void)application:(UIApplication *)application  
2.   didReceiveRemoteNotification:(NSDictionary *)userInfo  
3.   fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler
```

NSURLSession之BackgroundSession



这个主要就是用来处理大数据量的下载的，其保证应用即使在后台也不影响数据的上传和下载。用法：

- 1.创建NSURLSession的backgroundSession，并对其进行配置（参见上一篇文章）
- 2.使用该Session启动一个数据传输任务。
- 3.在AppDelegate中实现方法告诉应用：

Object-c代码  

```
1. - (void)application:(UIApplication *)application handleEventsForBackgroundURLSession:(NSString *)identifier  
2.   completionHandler:(void (^)(void))completionHandler
```

- 4.在具体的ViewController中实现NSURLSessionTask的Delegate方法，根据任务完成或出错的情况对UI进行更新：

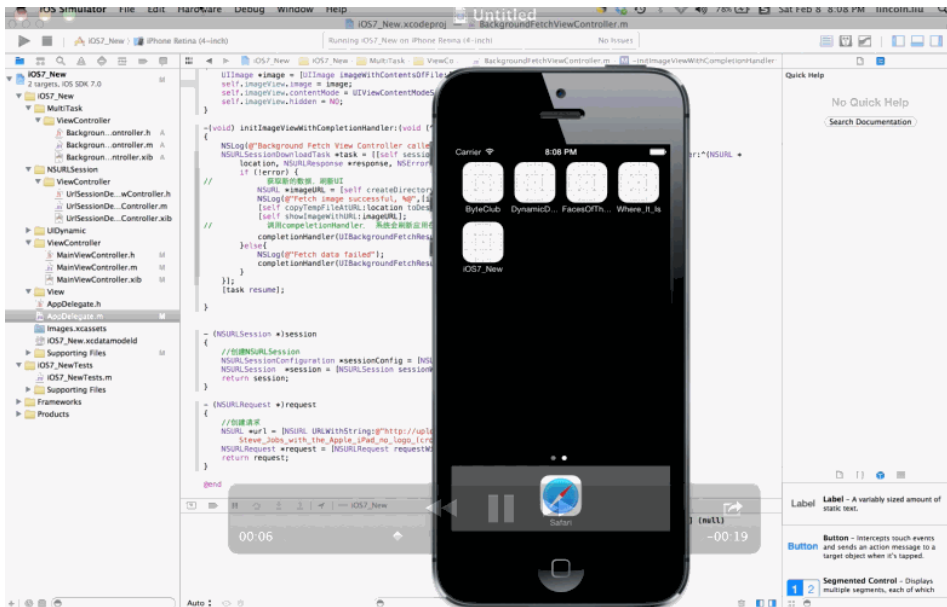
Object-c代码  

```
1. - (void)URLSession:(NSURLSession *)session downloadTask:(NSURLSessionDownloadTask *)downloadTask  
2.   didFinishDownloadingToURL:(NSURL *)location;  
3. - (void)URLSession:(NSURLSession *)session task:(NSURLSessionTask *)task  
4.   didCompleteWithError:(NSError *)error;
```

一旦后台传输的状态发生变化（包括正常结束和失败）的时候，应用将被唤醒并运行AppDelegate中的回调，接下来NSURLSessionTask的Delegate方法将在后台被调用

实战

这儿我实现了一个后台获取的Demo，我使用后台获取去获取一张图片，显示在页面上。下面是演示效果：



实现代码:
AppDelegate.m
Object-c代码

```
1. -(BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
2. {
3.     //设定BackgroundFetch的频率
4.     [[UIApplication sharedApplication] setMinimumBackgroundFetchInterval:UIApplicationBackgroundFetchIntervalMinimum];
5.     ...
6. }
7.
8. #pragma mark - backgroundFetch delegates
9. -(void) application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler
10. {
11.     //当后台获取被触发时，或调用下面的代码，这儿我直接去调用MainViewController的方法去刷新UI。
12.     UINavigationController *navigationController = (UINavigationController *)self.window.rootViewController;
13.     MainViewController *mainViewController = (MainViewController *)navigationController.topViewController;
14.     [mainViewController fetchImageWithCompletionHandler:completionHandler];
15.     //修改Icon的badgeNumber提醒后台又刷新，强迫症者慎用
16.     [UIApplication sharedApplication].applicationIconBadgeNumber += 1;
17. }
```

MainViewController.m
Object-c代码

```
1. -(void)fetchImageWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler
2. {
3.     //调用子UIViewController的方法刷新UI
4.     NSLog(@"main controller called fetch");
5.     [self.backgroundFetchViewController initImageViewWithCompletionHandler:completionHandler];
6. }
```

BackgroundFetchViewController.m
Object-c代码

```
1. #import "BackgroundFetchViewController.h"
2.
3. @interface BackgroundFetchViewController ()
4.
5. @end
6.
7. @implementation BackgroundFetchViewController
8.
9. -(id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
10. {
11.     self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
12.     return self;
13. }
14.
15. -(void)viewDidLoad
```

```
16. {
17.     [super viewDidLoad];
18. }
19.
20. - (void)didReceiveMemoryWarning
21. {
22.     [super didReceiveMemoryWarning];
23. }
24.
25. -(NSURL *)createDirectoryForDownloadItemFromURL:(NSURL *)location
26. {
27.     NSFileManager *fileManager = [NSFileManager defaultManager];
28.     NSArray *urls = [fileManager URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask];
29.     NSURL *documentsDirectory = urls[0];
30.     return [documentsDirectory URLByAppendingPathComponent:[location lastPathComponent]];
31. }
32.
33. //把文件拷贝到指定路径
34. -(BOOL) copyTempFileAtURL:(NSURL *)location toDestination:(NSURL *)destination
35. {
36.
37.     NSError *error;
38.     NSFileManager *fileManager = [NSFileManager defaultManager];
39.     [fileManager removeItemAtURL:destination error:NULL];
40.     [fileManager copyItemAtURL:location toURL:destination error:&error];
41.     if (error == nil) {
42.         return true;
43.     }else{
44.         NSLog(@"%@",error);
45.         return false;
46.     }
47. }
48.
49. - (void)showImageWithURL:(NSURL *)imageURL
50. {
51.     UIImage *image = [UIImage imageWithContentsOfFile:[imageURL path]];
52.     self.imageView.image = image;
53.     self.imageView.contentMode = UIViewContentModeScaleAspectFit;
54.     self.imageView.hidden = NO;
55. }
56.
57. -(void) initImageViewWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler
58. {
59.     NSLog(@"Background Fetch View Controller called");
60.     NSURLSessionDownloadTask *task = [[self session] downloadTaskWithRequest:
[self request] completionHandler:^(NSURL *location, NSURLResponse *response, NSError *error) {
61.         if (!error) {
62.             // 获取新的数据，刷新UI
63.             NSURL *imageURL = [self createDirectoryForDownloadItemFromURL:location];
64.             NSLog(@"Fetch image successful, %@",[imageURL path]);
65.             [self copyTempFileAtURL:location toDestination:imageURL];
66.             [self showImageWithURL:imageURL];
67.             // 调用completionHandler， 系统会刷新应用在App Switcher中的UI
68.             completionHandler(UIBackgroundFetchResultNewData);
69.         }else{
70.             NSLog(@"Fetch data failed");
71.             completionHandler(UIBackgroundFetchResultFailed);
72.         }
73.     }];
74.     [task resume];
75. }
76. }
77.
78.
79. - (NSURLSession *)session
80. {
81.     //创建NSURLSession
82.     NSURLSessionConfiguration *sessionConfig = [NSURLSessionConfiguration defaultSessionConfiguration];
83.     NSURLSession *session = [NSURLSession sessionWithConfiguration:sessionConfig];
84.     return session;
85. }
86.
87. - (NSURLRequest *)request
88. {
89.     //创建请求
90.     NSURL *url = [NSURL URLWithString:@"http://upload.wikimedia.org/wikipedia/commons/e/e0/Steve_Jobs_with_the_Apple_iPad_no_logo_(cropped).jpg"];
91.     NSURLRequest *request = [NSURLRequest requestWithURL:url];
92.     return
```

所有代码在这儿: https://github.com/xianlinbox/iOS7_New

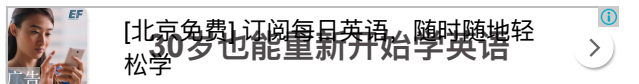
参考文章:

<http://www.shinobicontrols.com/blog/posts/2013/09/24/ios7-day-by-day-day-3-background-fetch>

<http://www.objc.io/issue-5/multitasking.html>

<http://onevc.com/2013/08/ios7-background-multitask/>

- [查看图片附件](#)





2

顶

0

踩

分享到:  

[Shell文件处理、Tomcat默认项目、AccessLo ...](#) | [实战iOS7之NSURLSession](#)

- 2014-02-08 20:20
- 浏览 9194
- [评论\(0\)](#)
- 分类: [移动开发](#)
- [相关推荐](#)

参考知识库

iOS

[iOS知识库](#) 282 关注 / 959 收录

评论

发表评论



[您还没有登录,请您登录后再发表评论](#)



ningandjin

- 浏览: 576787 次
- 性别: 
- 来自: 成都
-  我现在离线

最近访客 [更多访客>>](#)



[zfzone](#)



[tangrui9105](#)



[nuli888](#)



[abortexit](#)

文章分类

- [全部博客 \(122\)](#)
- [个人心情 \(1\)](#)
- [Java学习 \(42\)](#)
- [Linux学习 \(4\)](#)
- [好文分享 \(6\)](#)
- [遇到过的技术问题 \(4\)](#)
- [Ruby学习 \(20\)](#)
- [小刘说网络编程 \(1\)](#)
- [移动开发 \(27\)](#)
- [读书笔记 \(14\)](#)
- [JavaScript \(6\)](#)

社区版块

- [我的资讯 \(0\)](#)
- [我的论坛 \(1\)](#)
- [我的问答 \(1\)](#)

存档分类

- [2015-07 \(2\)](#)
- [2015-06 \(1\)](#)
- [2015-02 \(1\)](#)
- [更多存档...](#)

最新评论

- [学途在线](#): gradle+maven+springboot实战课程观看地址 ...
[从Maven到Gradle](#)
- [lrz0534](#): 你好，我在maven上面下载了swagger-springmv ...
[Micro Service工具集之Swagger: 可测试的样式化API文档](#)
- [sqz10200](#): 博主写的不错，思想和写的代码都挺好 我平时都是看老外的 ...
[Spring MVC中的异常处理](#)
- [foreverwey](#): evan8126 写道不用搞的那么复杂，jenv很方便http ...
[Mac下同时安装多个版本的JDK](#)
- [baungham](#): qdujunjie 写道powertech 写道阿里的dubb ...
[使用Hystrix守护应用 \(1\)](#)

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。
© 2003-2016 ITeye.com. All rights reserved. [京ICP证110151号 京公网安备110105010620]



ios下载



ios在线培训



ios系统下载



教育学