

从零开始学iOS7开发系列教程-事务管理软件开发实战-Chapter12

版权声明：

原文及示例代码来自raywenderlich store中的iOS Apprentice 系列2教程，经过翻译和改编。

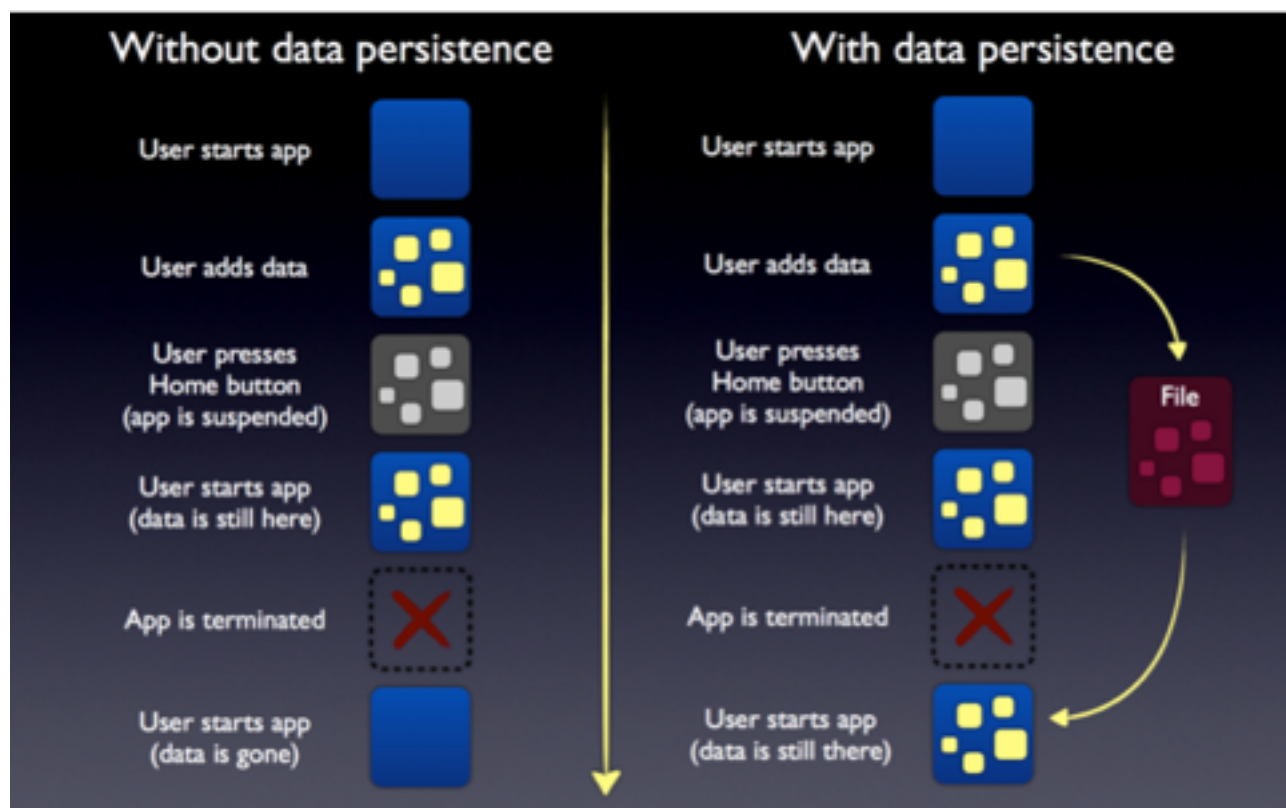
版权归原作者所有，本系列教程仅供学习参考使用，感兴趣的朋友建议购买原教程(<http://www.raywenderlich.com/store/ios-apprentice>)。

欢迎继续我们的学习。

回过头来看看，在短短11章的内容中，我们已经接触了iOS开发的很多重要知识点。相信在本系列教程完成后，我们就可以自豪的宣传自己对于iOS开发算是真正的入门了。

目前来说，对于用户新添加到列表中的事项，只要退出应用就会自动消失。我们也可以尝试删除或修改列表中的五个默认事项。不过一旦重新启动应用，我们所做的一切似乎从未发生过。显然真实世界中的应用不应该也不可能是这样来运作的。

感谢iOS新的多任务特性，现在当我们关闭某个应用的时候它会驻留在内存中。虽然它会进入某种类似休眠的状态，但所有的数据信息仍然保持原貌。在正常情况下，用户永远不会终止某个应用，而只是让它“悬停”在后台。不过在某些情况下，比如当iPhone内存不足的时候，iOS系统会自动清理掉后台那些还在休眠的应用，从而清理出一些内存出来。而且在某些情况下，只要用户愿意，他们可以手动kill掉后台中的应用，或者干脆reset整个设备。



显然将待办事项列表保留在内存中并不足以让它们永远活下去。因此，我们需要将数据保存在iPhone的闪存存储空间的文件中。这就好比你在电脑上写文章的时候过一会儿就要保存一下，只不过iPhone应用应该具备自动保存的功能。没必要为此去麻烦用户触碰某个Save按钮。

在上图中，左侧是没有数据持久保存的情况下，当某种情况（内存不足，用户手动kill掉后台的应用等）发生时，用户所添加的数据会消失无影踪。而右侧则是有数据持久保存的情况，无论何时，用户所添加的数据都会完好无损。

在接下来的这部分内容中，我们将要学习：

- 1.为了能让代办事项清单中的事项可以持久保存，判断该把保存这些信息的文件放到文件系统的何处。
- 2.当用户更改代办事项时（如添加一个新项目，开启/关闭勾选标志等），将待办事项保存到文件中。
- 3.当应用重新启动时从该文件加载代办事项清单。

好了，目标已定，接下来就是该如何具体实现了。

iOS中的文件目录

iOS应用居住在一个与世隔绝的桃花源中，世人称之为“沙盒”。简单来说，每个应用都有自己的特定目录用来保存文件和各类信息，但却不能访问其它应用的目录或文件。想象一个庞大无比的虚拟世界，在这个世界中所有的居民都生活在一个名为“沙盒”的子系统。作为这个世界的一员，你可以通过沙盒和母系统进行交互，但却不能和任何生活在其它“沙盒”中的居民直接交流。这个貌似颇有些Matrix（黑客帝国）的影子。苹果之所以这样设计并非因为变态，而是为了安全考虑。通过这种沙盒机制，可以有效防止恶意软件或者病毒的侵袭。相比Android系统来说，iOS系统在安全性能方面的确要技高一筹。因为任何一个应用只能破坏沙盒中属于自己的文件，就不可能破坏系统其它的部分。

在iOS中，应用可以将文件保存在所谓的“documents”目录中。而这个目录将永远生活在应用的沙盒之中。当用户使用iTunes或iCloud备份时，Documents文件中的文件也会被备份。当开发者发布了一个新版本时，用户会安装更新内容，但Documents文件夹却依然完好无损。因此任何保存在该文件中的数据可谓延年益寿不必时时刻刻战战兢兢被删除。

好吧，理论知识就这些，让我们看看如何具体操作。

打开Xcode,切换到ChecklistsViewController.m，然后在viewDidLoad方法之后添加以下方法：

```
-(NSString*)documentsDirectory{

    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);
    NSString *documentsDirectory = [paths firstObject];

    return documentsDirectory;
}

-(NSString*)dataFilePath{

    return [[self documentsDirectory]stringByAppendingPathComponent:@"Checklists.plist"];
}
```

这里的documentsDirectory是哥自己添加的一个方法。目前来说并没有一个标准方法可以直接调用来获取到Documents 文件夹的完整路径，于是哥只好费力自己创建了一个。当然后面你可以直接调用它来获取到Documents文件夹的完整路径，为此是不是应该感谢哥？

dataFilePath方法调用了documentsDirectory方法来创建到该文件的完整路径，我们将使用该文件来保存checklist项目。该文件的名称是Checklists.plist，而它就住在Documents文件夹里面。

这里我们使用NSString的stringByAppendingPathComponent方法来创建到Checklists.plist文件的完整系统路径。之所以可以直接调用这个方法，是因为[self documentsDirectory]返回的是一个NSString对象。别忘了NSString对象是immutable（不可变的），因此该方法创建了一个新的字符串对象，也就是到目标文件的最终完整路径。

当然，我们也可以用类似下面的方式来获取完整路径：

```
- (NSString *)dataFilePath {  
    return [NSString stringWithFormat:@"%s/%Checklists.plist", [self documentsDirectory]];  
}
```

在以上方法的代码中，我们将“Checklists.plist”添加到Documents的文件目录中。不过我个人仍然推荐使用stringByAppendingPathComponent方法，因为这样会减少出错的可能。

接下来在viewDidLoad方法中添加两个NSLog()语句：

```
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
  
    NSLog(@"文件夹的目录是： %@",[self documentsDirectory]);  
    NSLog(@"数据文件的最终路径是： %@",[self dataFilePath]);  
  
    _items = [[NSMutableArray alloc] initWithCapacity:20];  
  
    ChecklistItem *item;  
  
    item = [[ChecklistItem alloc] init];  
    item.text = @"观看嫦娥飞天和玉兔升空的视频";  
    item.checked = NO;  
    [_items addObject:item];  
  
    item = [[ChecklistItem alloc] init];  
    item.text = @"了解Sony a7和MBP的最新价格";  
    item.checked = NO;  
    [_items addObject:item];  
  
    item = [[ChecklistItem alloc] init];
```

```

item.text=@"复习苍老师的经典视频教程";
item.checked = NO;
[_items addObject:item];

item = [[ChecklistItem alloc]init];
item.text=@"去电影院看地心引力";
item.checked = NO;
[_items addObject:item];

item = [[ChecklistItem alloc]init];
item.text=@"看西甲巴萨新败的比赛回放";
item.checked = NO;
[_items addObject:item];

item = [[ChecklistItem alloc]init];
item.text=@"去香天下吃首相套餐";
item.checked = NO;
[_items addObject:item];

}

```

好了，现在编译运行项目，在Xcode的console部分会显示目录的真实地址。

假定我们从Simulator里面运行应用，就会看到类似下面的显示：

```

2013-12-15 19:54:21.134 Checklists[961:70b] 文件夹的目录是： /Users/happybubsy/Library/
Application Support/iPhone Simulator/7.1-64/Applications/B157795E-E647-4F48-8F12-
E46A8E737385/Documents
2013-12-15 19:54:21.142 Checklists[961:70b] 数据文件的最终路径是： /Users/happybubsy/
Library/Application Support/iPhone Simulator/7.1-64/Applications/B157795E-E647-4F48-8F12-
E46A8E737385/DocumentsChecklists.plist

```

加入我们在iPhone设备上测试应用，会看到类似下面的显示：

```

2013-12-15 19:58:50.709 Checklists[589:60b] 文件夹的目录是： /var/mobile/Applications/
80406540-43FC-42BE-BE92-94CC7A040476/Documents
2013-12-15 19:58:50.717 Checklists[589:60b] 数据文件的最终路径是： /var/mobile/
Applications/80406540-43FC-42BE-BE92-94CC7A040476/Documents/Checklists.plist

```

```

2013-12-15 19:58:50.709 Checklists[589:60b] 文件夹的目录是： /var/mobile/Applications/
80406540-43FC-42BE-BE92-94CC7A040476/Documents
2013-12-15 19:58:50.717 Checklists[589:60b] 数据文件的最终路径是： /var/mobile/
Applications/80406540-43FC-42BE-BE92-94CC7A040476/Documents/Checklists.plist

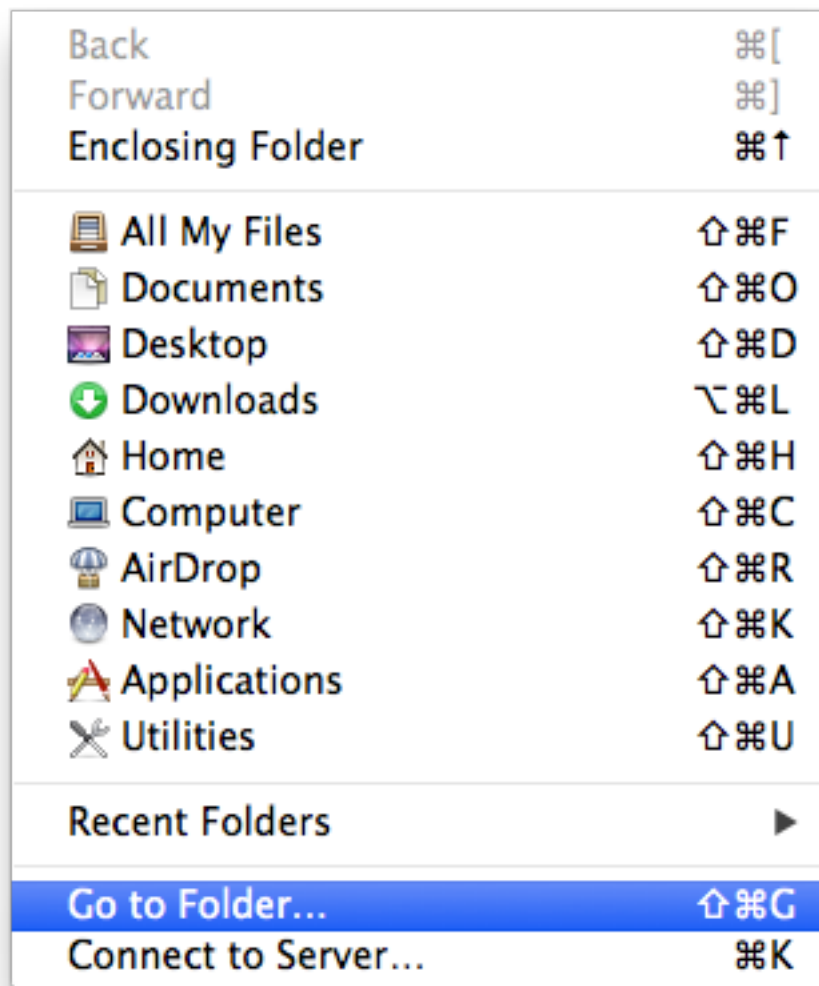
```

注意到在Simulator上该应用的目录是 B157795E-E647-4F48-8F12-E46A8E737385/, 而在设备上的目录是 80406540-43FC-42BE-BE92-94CC7A040476/

其实上面的字母和数字组合是Xcode (或iTunes)在Simulator或设备上安装时随机生成的一个ID.也就是说该目录其实就是应用的沙盒。

为了简便起见, 在本部分内容的后续部分我们只在Simulator中测试。因为Simulator会将应用的相关文件都保存在Mac上的固定位置, 我们可以通过Finder来随时查看其中的内容。

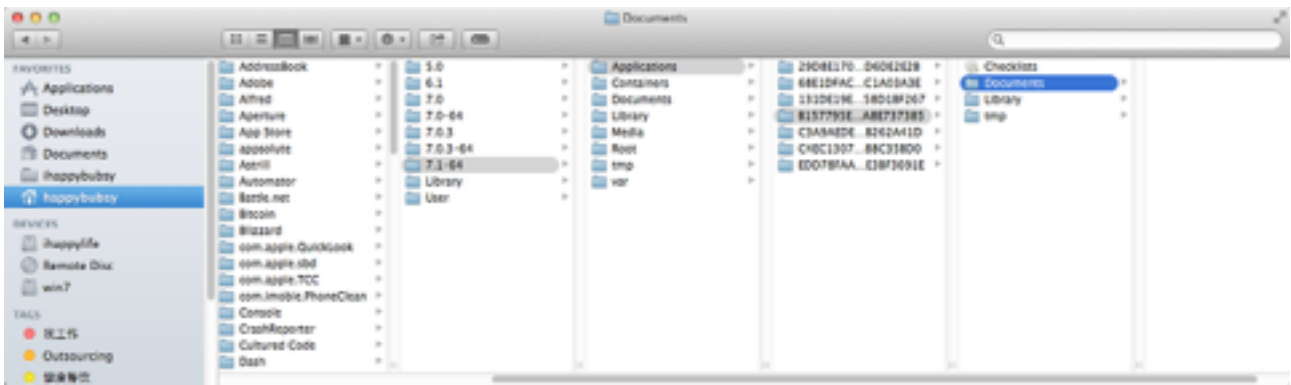
好了, 在Mac系统中打开Finder, 然后从菜单中选择Go-Go to Folder



然后将刚才的
贴到对话框中,
贴的就是:

完整目录拷贝粘
比如这里我们粘

/Users/happybubsy/Library/Application Support/iPhone Simulator/7.1-64/Applications/
B157795E-E647-4F48-8F12-E46A8E737385/Documents
此时就会切换到对应的目录。

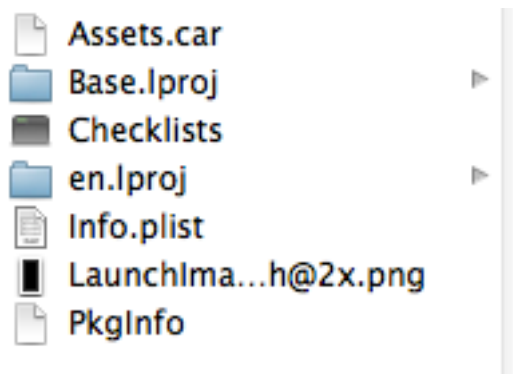


现在我们保持这个目录处于打开状态，然后看看后面创建Checklists.plist文件的时候，就会在这个目录中看到该文件的出现。

当然，如果你仔细观察，还会看到这个目录中的其它内容：

- 1.Checklists.app，也就是我们的application bundle.
- 2.Documents文件夹，也就是应用保存数据文件的目录。当前其中的内容是空白的
- 3.Library文件夹，其中会存放一些缓存文件和偏好设置文件。其中的内容都是由操作系统亲自管理的，亲你还是别想太多了。
- 4.tmp文件夹，用于保存临时文件。有时应用会需要创建一些临时用途的文件，通常就会保存在这个临时文件夹中。操作系统会时不时清理这个文件夹中的内容。

好吧，我猜你回对application bundle（应用束）中的内容感到好奇，只需要右键单击，然后选择Show Package Contents，就可以看到其中的内容了。



APP bundle其实是一个文件夹，不过我们伟大的Finder会假装它只是个文件而已。通过Show Package Contents,我们就会看到这个文件夹中的详细内容。

好吧，看起来有点高深莫测，我们先别管里面的内容，在后续的教程中会对此进行详细解释的。

当然，其实我们也可以在设备上直接查看应用的Documents文件夹。在iPhone或iPad中，切换到Settings-General-Usage,然后触碰某个应用的名称，就会看到documents文件夹中的内容。



好了，现在我们已经知道这个沙盒的真实地址在哪里了。那么，接下来该如何保存checklist代办事项呢？

预知后事如何且听下回分解。

今天的学习到此为止，还是送上点福利吧。

首先恭贺恒大队力克埃及法老！！



接下来当然还是看美女了。



