

从零开始学iOS7开发系列教程-事务管理软件开发实战-Chapter21

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。

开发环境：

Xcode 5 +iOS 7

在上一篇的内容中，我们初次接触了NSUserDefaults。这里我们来尝试继续使用它。

当用户首次打开应用的时候，如果可以看到一个默认的checklist（命名为List），并且直接切换到这个界面，岂非秒事？用雷布斯、周大炮和乔帮主的话来说，这就是所谓的超出用户期望。在设计开发iOS的时候，永远记住要为用户多考虑一点点。也许就是这一点点的用户体验提升就可以让你的应用受到用户的青睐。

通过设置一个默认的checklist，可以让用户立即开始添加代办事项。实际上苹果的Notes应用就是这样工作的：即便是首次打开应用，用户也可以立即开始记录笔记，当然你也可以选择通过导航层级返回到所有笔记本的列表界面。

为了实现这个小小的突破，我们需要用到NSUserDefaults来记录相关信息，判断用户是否是首次打开应用。如果是，就创建一个新的Checklist对象。我们还是在DataModel中实现所有的相关逻辑。

打开Xcode，切换到DataModel.m，然后在文件的顶部输入：

```
#import "Checklist.h"
```

因为之前我们已经有一个registerDefaults方法了，所以可以直接把新的默认设置保存在这里。

更改registerDefaults方法如下：

```
-(void)registerDefaults{  
  
    NSDictionary *dictionary = @{@"ChecklistIndex" :@-1,@"FirstTime":@YES};  
  
    [[NSUserDefaults standardUserDefaults]registerDefaults:dictionary];  
}
```

FirstTime是一个布尔类型的值，它的值只有两种可能。当用户首次安装应用时，“FirstTime”的值需要设置为YES。

理论知识充电：

基本数据类型 vs 对象

有一点需要特别注意的是，在iOS开发中，词典中不能保存基本数据类型，比如int和BOOL，而只能保存对象。对数组来说也是如此。如果我们希望将一个int或BOOL值放到词典或数组中，就必须将其转换成NSNumber对象。

之前我曾经提到过基本数据类型和对象的区别。在某些编程语言中，一切都是对象。而在Objective-C语言中，“几乎”一切都是对象。

当然，对于某些简单的数学运算，使用对象稍显得有点奢侈，此时用基本数据类型要更方便。

区分基本数据类型和对象很简单，只需要看名称前是否有*就好了。只有对象前面才需要添加星号。此外，我们不能使用[[alloc]init]来创建基本数据类型。

有时我们需要在基本数据类型和对象之间进行转换。为了将int 或者BOOL类型的值放入词典中，我们首先需要把它封装到一个NSNumber对象中。我们使用@value的形式来完成这项工作，比如@-1 或者@YES，或者调用方法[NSNumber numberWithInt:]或者[NSNumber numberWithBool:]。

还有其它的方式也可以实现这一点：比如为了从NSNumber中获取一个整数值，可以用[number intValue]的方式。对于BOOL类型的值则采用[number boolValue]。

好吧，也许你看了上面的这项内容后对基本数据类型和对象的区别还是不知所谓。不过也不用担心，下一系列的教程中将对此进行更为详细的解释。

ok,回到代码时间。

在DataModel.m中添加一个handleFirstTime的新方法：

```
-(void)handleFirstTime{

    BOOL firstTime = [[NSUserDefaults standardUserDefaults]boolForKey:@"FirstTime"];

    if(firstTime){

        Checklist *checklist = [[Checklist alloc]init];

        checklist.name = @"List";

        [self.lists addObject:checklist];

        [self setIndexofSelectedChecklist:0];

        [[NSUserDefaults standardUserDefaults]setBool:NO forKey:@"FirstTime"];

    }
}
```

这里我们检查了NSUserDefaults中“FirstTime”键的值。不过这里我们可以忽略NSNumber，然后从NSUserDefaults中直接获取一个布尔值。仅当我们需要注册初始值的时候才需要用到NSNumber（因为NSUserDefaults并非是一个真正的词典对象，只是看起来像）。

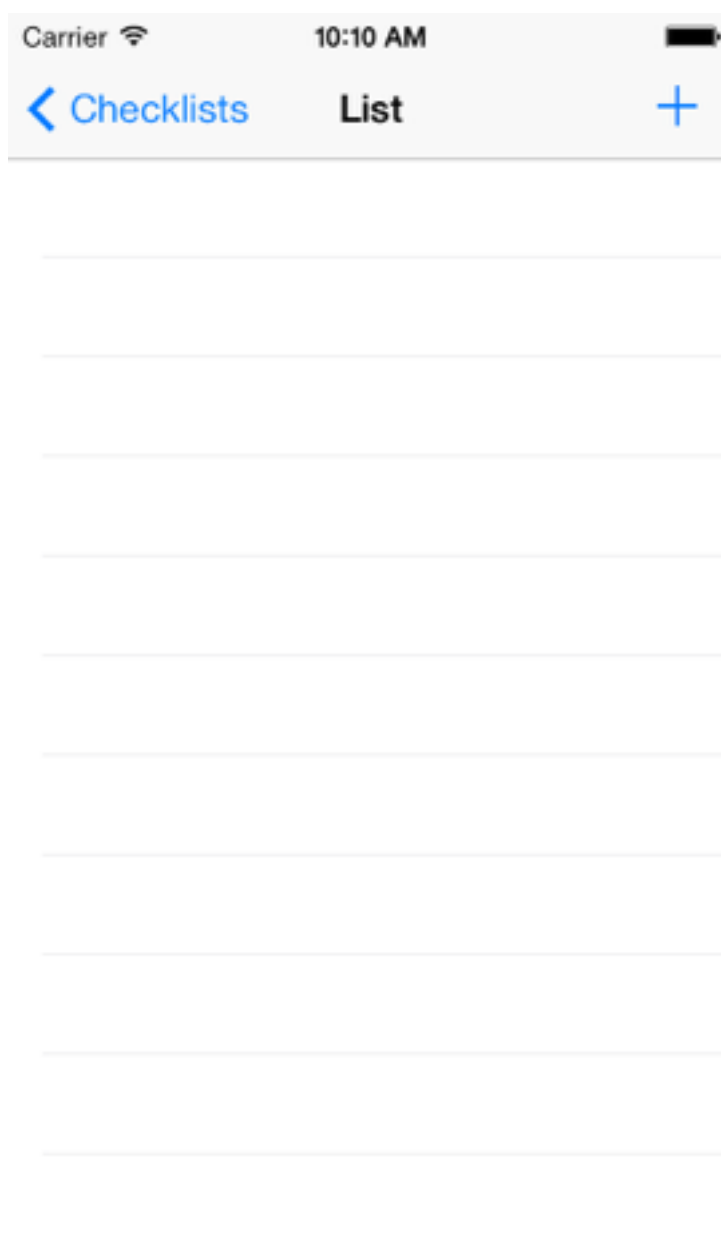
如果“FirstTime”的值是YES，就说明是应用首次运行。此时我们需要创建一个新的Checklist对象，并将其添加到数组中。我们还需要调用setIndexofSelectedChecklist来确保应用将会自动segue到

新的checklist（通过AllListsViewController的viewDidAppear方法）。最后，我们需要将FirstTime设置为NO，这样下次应用启动的时候就不会运行这部分代码。

接下来我们需要从init中调用这个新的方法：

```
-(id)init{  
  
    if((self =[super init])){  
  
        [self loadChecklists];  
        [self registerDefaults];  
        [self handleFirstTime];  
    }  
    return self;  
}
```

好了，现在从Simulator中删掉这个应用，然后从Xcode中再次编译运行。
因为是首次运行该应用，它将自动创建一个新的名为List的checklist，并切换到该界面。



在继续前进之前，让我们先稍微休息一下。从下一章开始，我们即将进入本系列教程的最后部分。

不管怎样，还是先送上今日福利，香车美女

