



# iOS 5 Programming Cookbook

## 第四章 Storyboards

版本 1.0

翻译时间：2012-05-28

DevDiv 翻译： kyelup cloudhsu 耐心摩卡  
wangli2003j3 xiebaochun

DevDiv 校对： laigb kyelup

DevDiv 编辑： BeyondVincent

## 写在前面

目前，移动开发被广大的开发者们看好，并大量的加入移动领域的开发。

鉴于以下原因：

- 国内的相关中文资料缺乏
- 许多开发者对 E 文很是感冒
- 电子版的文档利于技术传播和交流

[DevDiv.com](http://DevDiv.com) [移动开发论坛](#) 特此成立了翻译组，翻译组成员具有丰富的移动开发经验和英语翻译水平。组员们利用业余时间，把一些好的相关英文资料翻译成中文，为广大移动开发者尽一点绵薄之力，希望能对读者有些许作用，在此也感谢组员们的辛勤付出。

### 关于 DevDiv

DevDiv 已成长为国内最具人气的综合性移动开发社区

更多相关信息请访问 [DevDiv 移动开发论坛](#)。

### 技术支持

首先 DevDiv 翻译组对您能够阅读本文以及关注 DevDiv 表示由衷的感谢。

在您学习和开发过程中，或多或少会遇到一些问题。DevDiv 论坛集结了一流的移动专家，我们很乐意与您一起探讨移动开发。如果您有什么问题和需要技术支持的话，请访问 [DevDiv 移动开发论坛](#) 或者发送邮件到 [BeyondVincent@DevDiv.com](mailto:BeyondVincent@DevDiv.com)，我们将尽力所能及的帮助您。

### 关于本文的翻译

感谢 kyelup、cloudhsu、耐心摩卡、wangli2003j3 和 xiebaochun 对本文的翻译，同时非常感谢 laigb 和 kyelup 在百忙中抽出时间对翻译初稿的认真校验，指出了文章中的错误。才使本文与读者尽快见面。由于书稿内容多，我们的知识有限，尽管我们进行了细心的检查，但是还是会存在错误，这里恳请广大读者批评指正，并发送邮件至 [BeyondVincent@devdiv.com](mailto:BeyondVincent@devdiv.com)，在此我们表示衷心的感谢。

读者查看下面的帖子可以持续关注章节翻译更新情况

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译各章节汇总](#)

DevDiv 翻译

## 目录

写在前面	2
关于 DevDiv	2
技术支持	2
关于本文的翻译	2
目录	4
前言	5
第 1 章 基础入门	6
第 2 章 使用控制器和视图	7
第 3 章 构造和使用 Table View	8
第 4 章 Storyboards	9
4.0. 介绍	9
4.1. 用 Storyboard 创建一个项目	9
4.1.1. 问题	9
4.1.2. 方案	9
4.1.3. 讨论	9
4.1.4. 参考	11
4.2. 在 Storyboard 上增加一个导航控制器	11
4.2.1. 问题	11
4.2.2. 方案	12
4.2.3. 讨论	12
4.2.4. 参考	19
4.3. 将数据从一个屏幕传输到另一个屏幕	19
4.3.1. 问题	19
4.3.2. 方案	19
4.3.3. 讨论	19
4.3.4. 参考	21
4.4. 在一个已有项目中添加 Storyboard	21
4.4.1. 问题	21
4.4.2. 方案	21
4.4.3. 讨论	22
4.4.4. 参考	22

## 前言

参考帖子

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译 前言](#)

DevDiv 翻译

## 第 1 章 基础入门

参考帖子

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译 第一章 基础入门](#)

DevDiv 翻译

## 第 2 章 使用控制器和视图

参考帖子

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译 第二章 使用控制器和视图\(上\)](#)

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译 第二章 使用控制器和视图\(下\)](#)

DevDiv 翻译

## 第 3 章 构造和使用 Table View

参考帖子

[\[DevDiv 翻译\]iOS 5 Programming Cookbook 翻译 第三章 构造和使用 Table View](#)

DevDiv 翻译



## 第 4 章 Storyboards

### 4.0. 介绍

我们现在已经习惯使用视图控制器。为了拖拽视图控制器我们知道如何使用导航控制器。Apple 相信这一点会更容易做到，这就是整个故事背后的 Storyboards。Storyboarding 是一个定义了 App 中不同屏幕中建立联系的新方法。例如，假如一年前你在 App 上编码了 20 独立视图控制器，现在要看源代码的话你需要找出这些视图控制器之间的联系、当用户采取特定动作时什么视图控制器会被拖出。这一点会很难解决，特别是如果你没有记录下自己的代码情况下。Storyboard 可以解决这个问题。通过 Storyboards，你可以浏览/创建整个 APP 的 UI，可以在一个屏幕的不同视图控制器之前建立联系。问题迎刃而解。

为了能够利用 Storyboard，你需要熟悉界面生成器。这就是本章的主要内容，所以你不必担心。

在 Storyboard 中，一个屏幕的内容称为一个场景。iPhone 上一个场景可以看成视图控制器上的视图，在你把内容放到场景中时也展现给了用户。在 iPad 上，由于屏幕尺寸更大，更多的场景会同时展现给用户。Storyboarding 支持从一个场景转换到另一个场景，如同导航控制器把一个新的视图控制器推到另一个顶部一样，这是一种转换。另一种转换是一个 modal 视图控制器从屏幕的底端滑出临时填充屏幕。在 iPad 上，为了显示出在某时刻是主要输出，modal 屏幕出现在屏幕的中央，并且通常使屏幕其余部分暗淡下来。

### 4.1. 用 Storyboard 创建一个项目

#### 4.1.1. 问题

在 Xcode 中，你想创建一个使用 Storyboard 的项目

#### 4.1.2. 方案

在 New Project 设置对话框中选择使用 Storyboard:



图 4-1.New Project 对话框中使用 Storyboard 选项

#### 4.1.3. 讨论

如果你想创建一个项目，其中使用了 Storyboards，简单完成下面的步骤：

1、在 Xcode 中，选择 File，然后选择 New-New Project...

2、在 New Project 对话框中，确定 iOS 主要类别被选中，选择 iOS 下的 Application 子类。完成之后，在右侧选择 Single View Application 然后点击 Next，结果如图 4-2 所示：

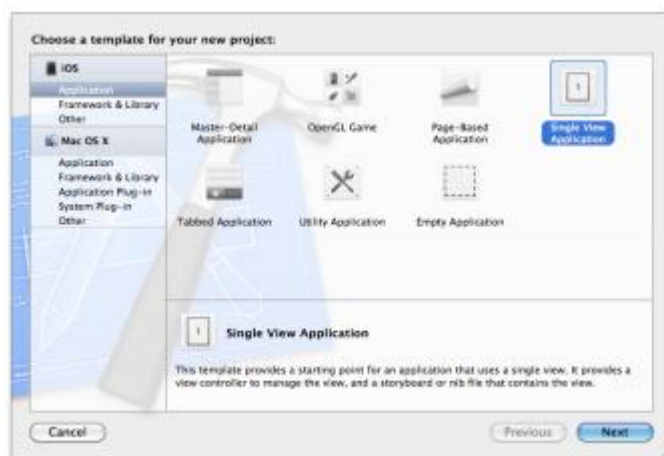


图 4-2. 为了使用 Storyboard 创建一个新 APP

3、现在选择一个产品名称，确保你的 APP 是一个通用 App。苹果想让开发者开始写通用 App，不仅能够满足 iPad 用户，也能使 iPhone 和 iPod Touch 的用户满意。在这个对话框中，确保你已经先查过 User Storyboards 的查验项，如图 4-3 所示；完成后请点击 Next：

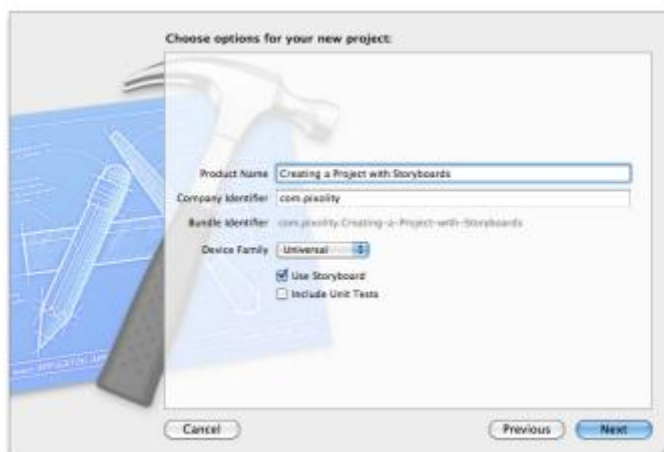


图 4-3. 为一个新项目使用 Storyboards

4、现在要求你把你把项目保存到一个文件夹中，保存之后，点击 Creat 键（见图 4-4），现在你有一个使用了 Storyboards 的项目。

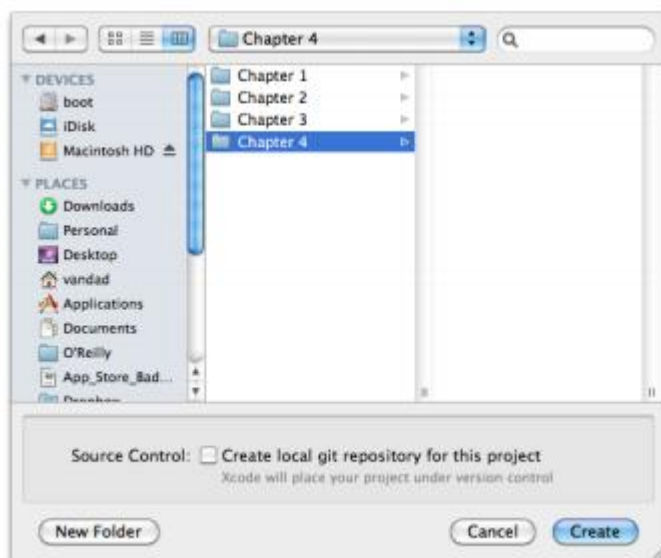


图 4-4. 把 Storyboard APP 保存到磁盘上

现在如果你看 Xcode 为你创建的文件夹(见图 4-5)，你会注意到有两个文件夹的名称以 .Storyboard. Xcode 结尾，因为这是一个通用 APP，分别为 iPhone 和 iPad 创建了一个 Storyboard，这样就可以决定两个设备上的想要的程序显示效果看起来一致。

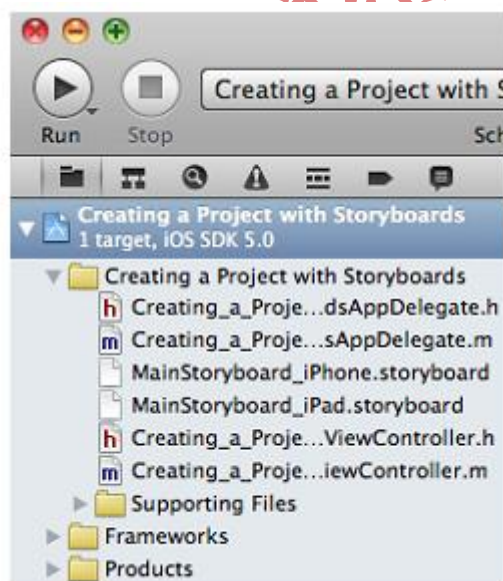


图 4-5. 两个 Storyboard 文件在一个通用 APP 中

#### 4. 1. 4. 参考 XXX

### 4. 2. 在 Storyboard 上增加一个导航控制器

#### 4. 2. 1. 问题

你想要能够在一个 Story-based APP 上管理多个视图控制器。

#### 4.2.2. 方案

创建一个导航控制器作为 Storyboard 文件夹的最初视图控制器。

#### 4.2.3. 讨论

如果你按照 4.1 中的方法，现在在 iPhone 虚拟机上运行 APP，你会注意到在屏幕顶部没有导航栏，有点像是在看一个白屏。这个问题背后的原因是，我们 Storyboard 文件的最初的视图控制器是一个视图控制器，它与导航控制器相对应。为了给 Story-based APP 增加一个导航控制器，简单操作下面几步：

1、点击 Xcode 为你创建的 iPhone Storyboard，我把它命名为我的项目-Adding a Navigation Bar to a Storyboard，所以 iPhone Storyboard 文件是 MainStoryboard\_iPhone.Storyboard。一旦你点击这个文件夹，界面生成器将自动展示这个文件夹的内容。

2、一旦 Storyboard 文件在 IB（界面生成器）中打开，只需双击一个空序列图像板画布上的空间，你会看到内容缩小，给你更多的自由发挥的空间，效果如图 4-6 看到的那样：

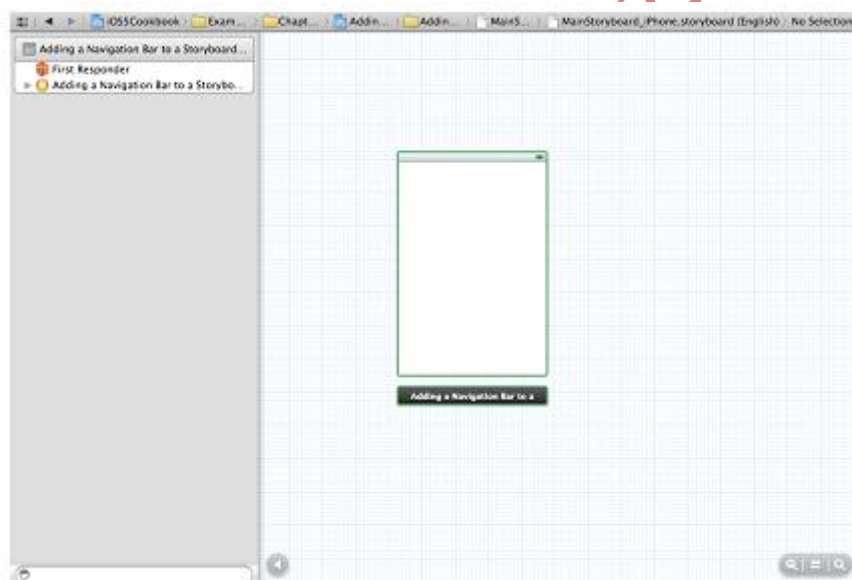


图 4-6. iPhone Storyboard 缩放视图

3. 在视图菜单下，选择 Utilities（实用程序）→ Show Object Library（显示对象库）。

4. 在对象库中，找到导航控制对象（见图 4-7），将其拖出后放到 Storyboard，放到现有的视图控制器的左侧（图 4-6），现在将看到与图 4-8 显示内容类似的情况。



图 4-7. 导航控制器对象在对象库中

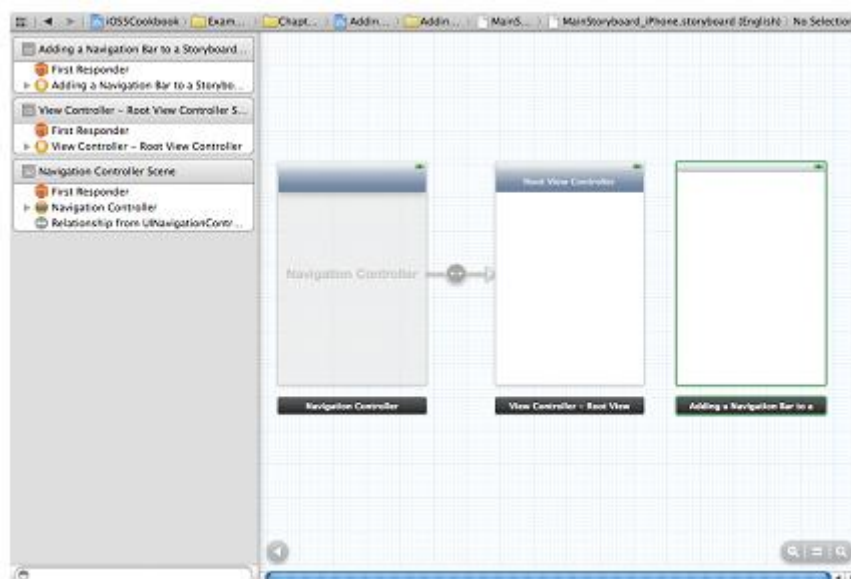


图 4-8. 一个导航控制器来自它自己的根视图控制器

1、和你在图 4-8 中看到的那样，导航控制器现在给我们的 UI 增加了另一个视图控制器，我们需要做的就是先选中它然后在键盘上点击删除键将其删除。现在我们只留下了导航控制器和最初的视图控制器，效果如图 4-9 所示。



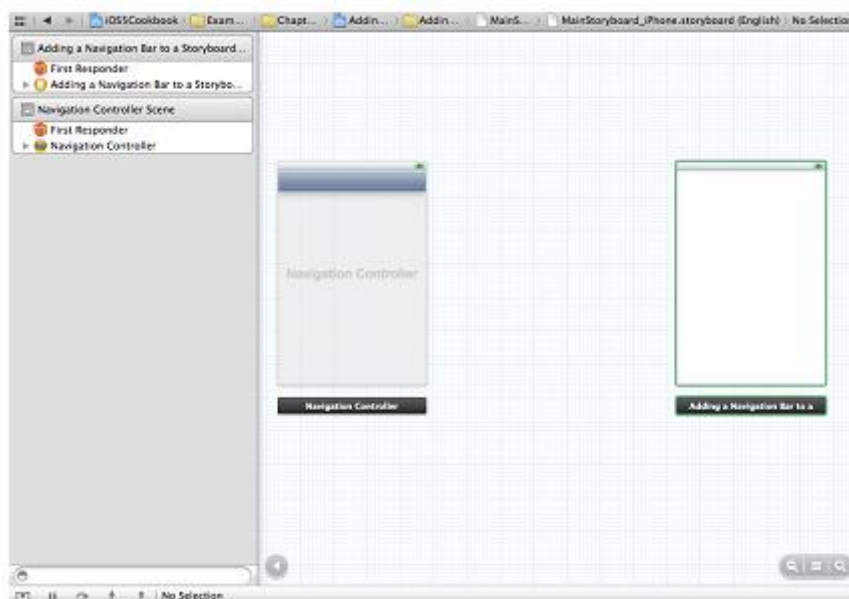


图 4-9. 删除来自导航控制器对象的根视图控制器

2、现在只要在 Storyboard 上点击导航控制器对象，一旦导航控制器被选中，选择键盘上的 Control 键和鼠标左键，将鼠标拖到我们最初在 Storyboard 的视图控制器（右侧）。这样就会从导航控制器到视图控制器中间画一条线。效果可以在图 4-10 中看到。

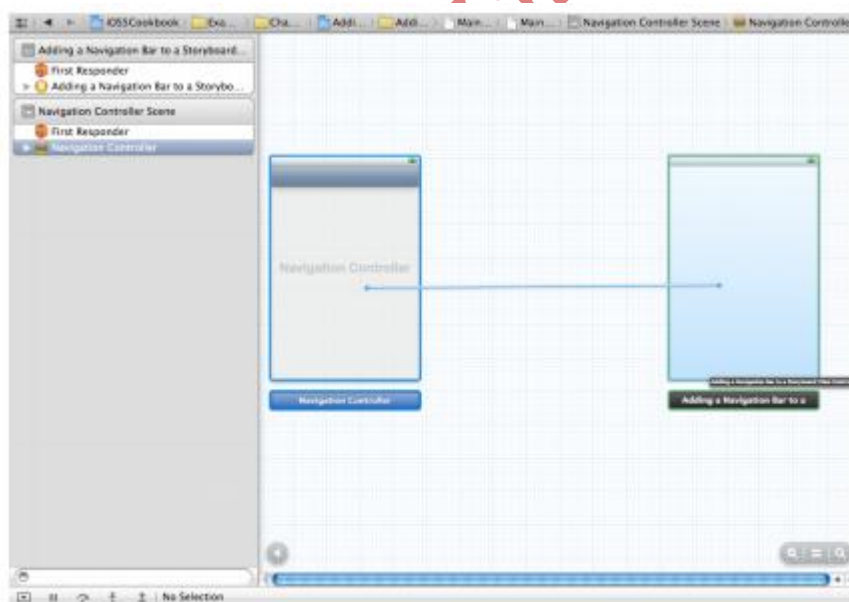


图 4-10. 连接导航控制器和最初的视图控制器

3、现在释放鼠标你会看到一个对话框提问，你想在导航和视图控制器之间建立哪类联系，从选项中选择 rootViewController 然后点击（见图 4-11）

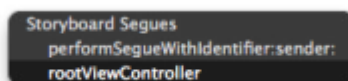


图 4-11. 将一个视图控制器设置为一个导航控制器的根

4、完成操作之后，Storyboard 会显示你的导航控制器已经和原始视图控制器连接起来，参看图 4-12：

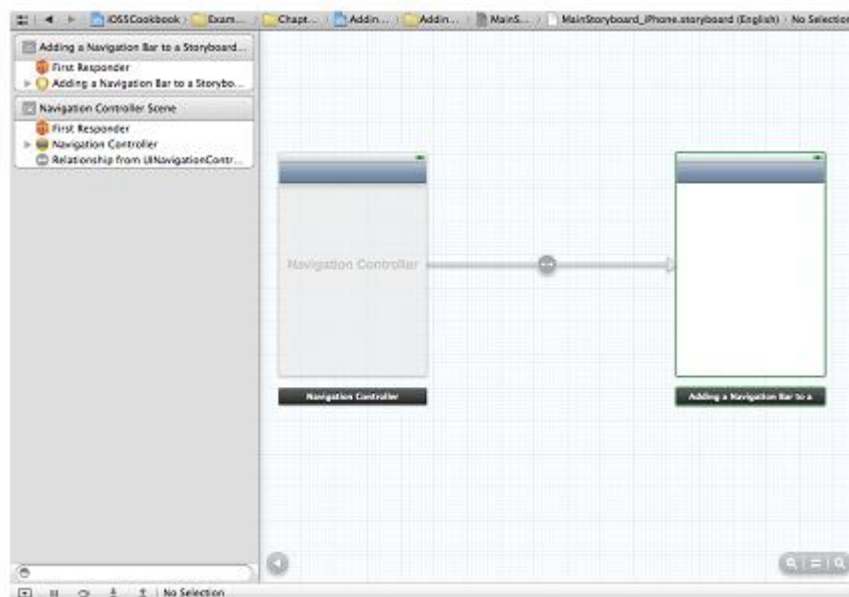


图 4-12. 导航控制器现在和最初视图控制器连接起来。

5、最后一步也是最重要的一步就是把导航控制器设置成初始/根视图控制器。如果不做这个，Storyboard 会把我们已有的初始视图控制器当成初始视图控制器来使用。在看下图 4-12，可以看到右侧的视图控制器有个边框。这表示了一个初始视图控制器。要把导航控制器转换成初始视图控制器，在界面生成器中的导航控制器场景面板下选择导航控制器就可以做到，如图 4-13 所示。现在在 Xcode 中点击 View 菜单，选择 View→Show Attributes Inspector(显示属性检查器)。只要属性检查器在视图控制器类中打开，就检查 Is Initial View Controller 检验箱（见图 4-14）。

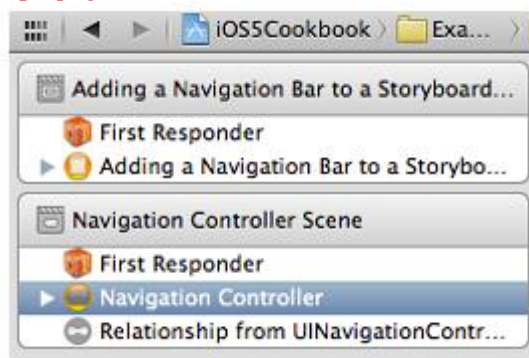


图 4-13. 在界面生成器中选择导航控制器

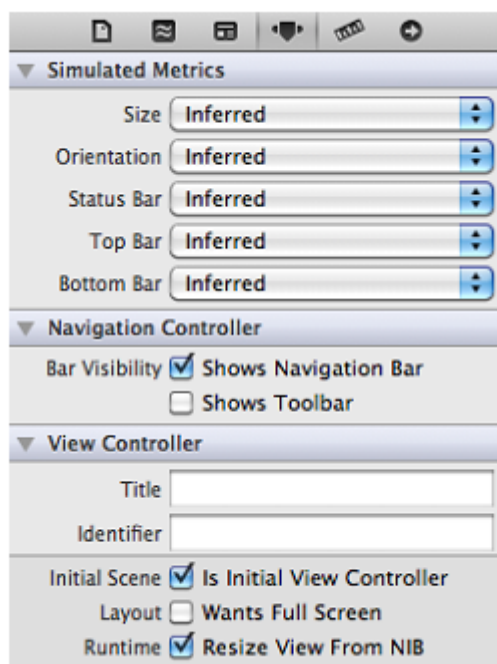


图 4-14. 选择一个导航控制器作为一个 Storyboard 的初始视图控制器

你可以看到，现在你的导航控制器有一个边框。现在如果你运行 APP，你会注意到初始视图控制器顶部有一个导航栏，说明这个视图控制器现在有一个导航控制器。在下一个方法中我们会看到如何使用导航控制器在屏幕上展示新场景。





图 4-15. 导航栏在一个通过 Storyboard 创建的视图控制器上

我们现在有一个带有导航控制器的视图控制器，但是我们的对象现在触发了一个动作，它从一个视图控制器移动到了另一个。那么，我们在视图控制器上放一个按键，只要用户按下这个按键，就会把一个视图控制器推出堆栈。听起来不错？那么我们开始：

1. 返回 Storyboard 文件

2. 从对象库中找出一个视图控制器对象，拖出后放到 Storyboard 上我们已存在的视图控制器的右侧，如图 4-17 中所看到的。



图 4-16. 对象库中的一个视图控制对象

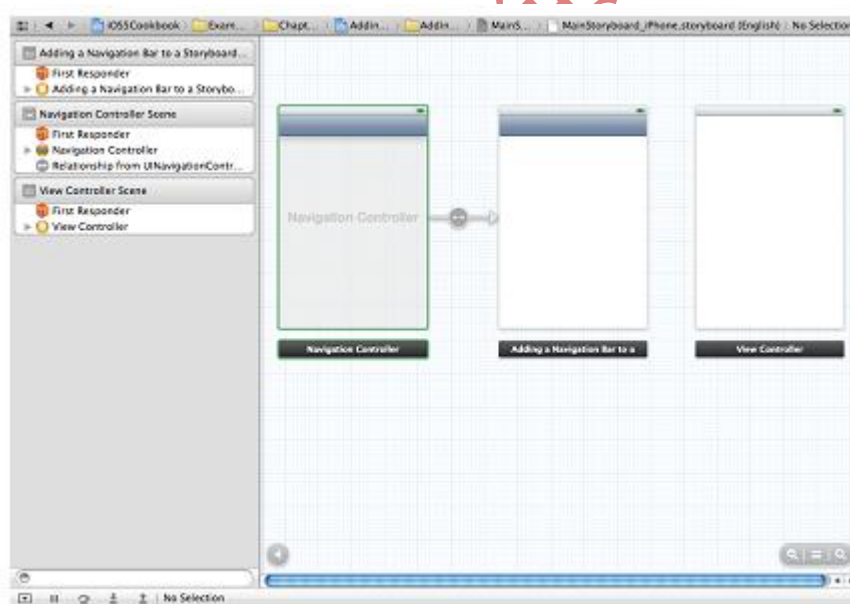


图 4-17. 在 Storyboard 上增加一个新的视图控制器

3. 在对象库中，找出按钮对象（见图 4-18），把它拖出对象库然后放入第一个视图控制器(见图 4-19)。注意如果你缩放试图控制器，界面生成器不允许你在视图控制器上增加按钮。在界面生成器允许你把 UI 组件放入视图控制器之前，你需要双击 Storyboard 的空置区域去缩放。



图 4-18. 在对象库中选择一个按钮对象

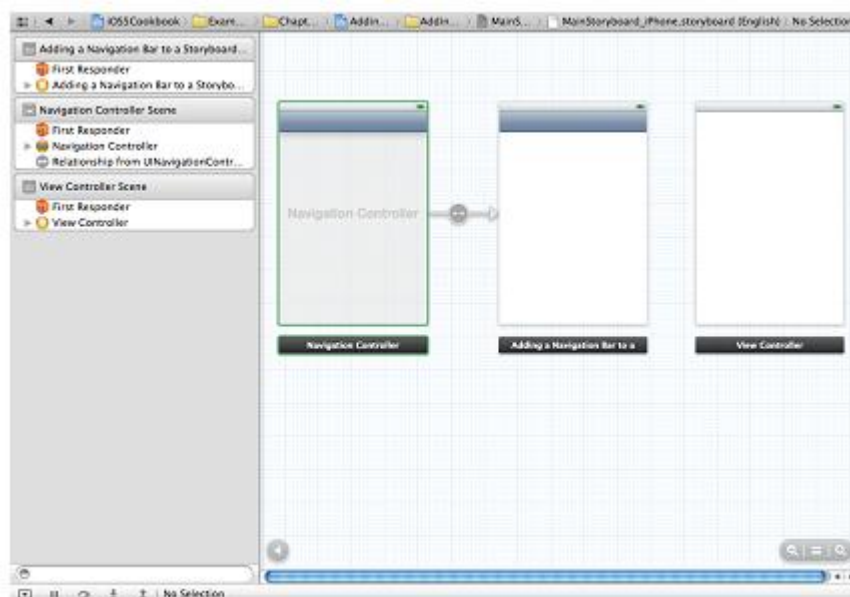


图 4-19. 把按钮放入 Storyboard 的第一个视图控制器

4. 然后选择按钮，按住键盘上的 **Control** 键和鼠标左键和按钮，一直拖到第二个视图控制器。(见图 4-20).

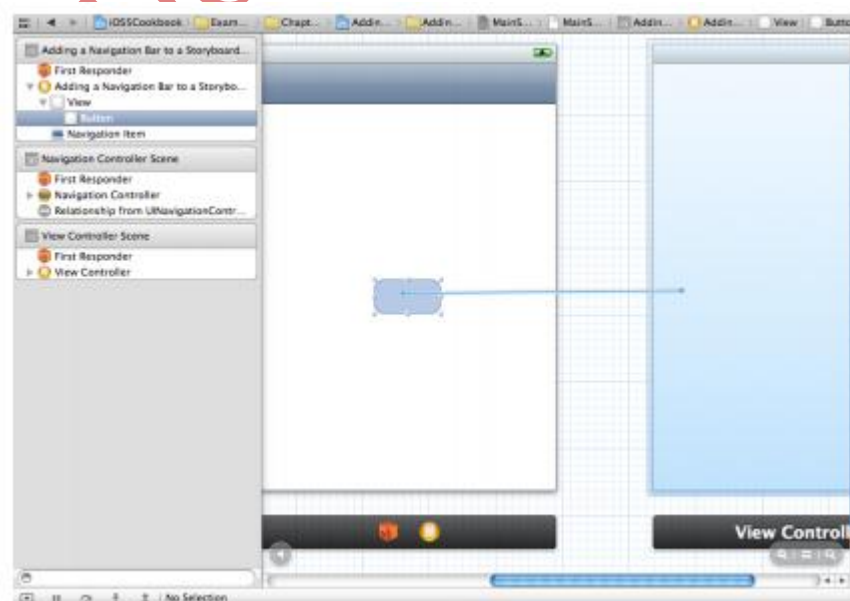


图 4-20.在 Storyboard 连接按钮和另一个视图控制器

5.现在松开鼠标左键和和键盘上的 Control 键，你会看到一个和图 4-21 类似的对话框，点击 performSegueWithIdentifier:sender:选项。

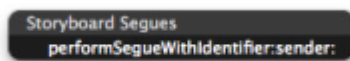


图 4-21. 使按钮成为 Segue

现在如果你看一眼 Storyboard，会看到第一个视图控制器和第二个视图控制器连起来了，见图 4-22：

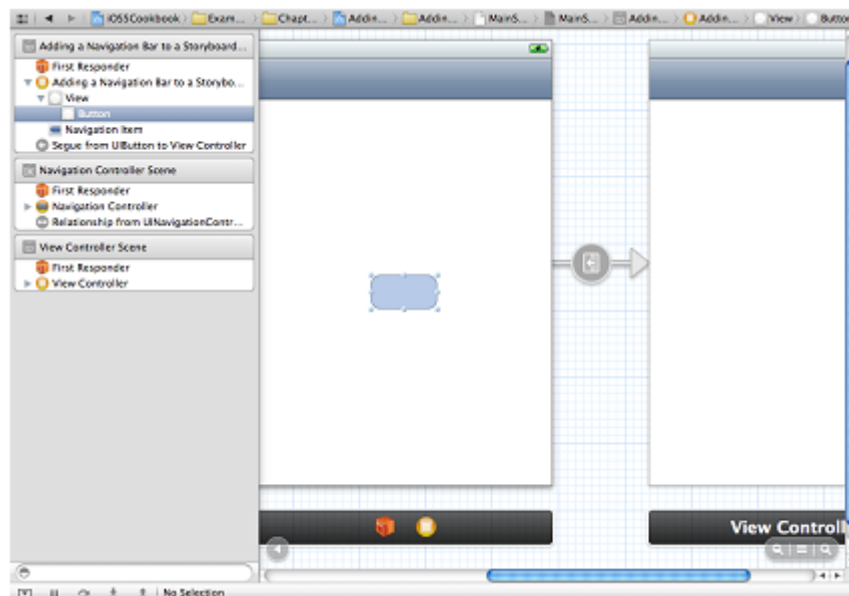


图 4-22. 第一个视图控制器通过一个 Segue 和第二个视图控制器连接

如果你运行 APP，点击第一个视图控制器的按钮，会看到第二个视图控制器自动从视图控制器堆栈拖出。只要第二个视图控制器展示出来，会看到导航杆上有个后退按钮。如果点击这个按钮，你就会返回到第一个视图控制器。

#### 4. 2. 4. 参考 XXX

### 4. 3. 将数据从一个屏幕传输到另一个屏幕

#### 4. 3. 1. 问题

想要使用 Storyboard 把数据从一个屏幕传到另一个屏幕。

#### 4. 3. 2. 方案 使用 Segue 对象。

#### 4. 3. 3. 讨论

和 Objective-C 中的其他对象一样，Segue 是一个对象。当发生不同场景之间的转换时，Storyboard 的运行库会为此转换创建一个 Segue 对象。Segue 就是类 UIStoryboardSegue 的

一个实例。当转换发生时，当前视图控制器（在 `Segue` 后会退出屏幕）将会收到 `prepareForSegue:sender:` 的消息，此处的 `prepareForSegue` 参数会变成 `UIStoryboardSegue` 类的一个对象。如果你想把当前视图控制器的数据传输到将要出现的视图控制器上，你需要使用 `prepareForSegue:sender:` 的方法。



为了能更好的理解这个方法，你要按照 4.2 中的指导在你的 Storyboard 上的一个导航控制器上创建 2 个视图控制器。

让我们在第一个视图控制器中执行 `prepareForSegue:sender:` 方法：

```
- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {  
  
    NSLog(@"Source Controller = %@", [segue sourceViewController]);  
    NSLog(@"Destination Controller = %@", [segue destinationViewController]);  
    NSLog(@"Segue Identifier = %@", [segue identifier]);  
  
}
```

现在如果你运行 APP，会在控制台窗口看到结果。然而，如果你注意的话，标示符为空。每个 `Segue` 有一个标示符，这个标示符可以唯一识别这个 `Segue`。由于一个场景有不止一个与之相关的 `Segue`，对你来说有一些能随后在视图控制器中检测的 `Segue` 标示符会好一点，你就相应的采取行动了。

一个界面生成器的 `Segue` 对象是两个场景的连接器，此处是我第一个视图控制器和第二个视图控制器之间的 `Segue`：



图 4-23.在界面生成器中选择一个 `Segue` 对象

按找下面的步骤你的 `Segue` 就有一个标示符：

1. 通过点击在界面生成器上选择一个 `Segue`。
2. 从视图菜单选择 `Utilities`→`Show Attributes Inspector`。

3.在 `Attributes Inspector` 上的标识符文本字段中，简单写下你想要 `Segue` 本身携带的标识符。你会看到，当 `Storyboard` 运行库在当前视图控制器中调用 `prepareForSegue:sender:` 方法为它准备 `Segue` 时，目标视图控制器已经在 `Segue` 对象中被初始化。现在是时机把需求数据传输到目标视图控制器。你既可以在目标视图控制器的一个属性中直接设置数据，也可以在

这个视图控制器上调用方法来传输数据，这完全取决于你。在这段代码中，我的第二个视图控制器是类 `SecondViewController`，并且已经把 `SimpleSegueToSecondViewController` 的标识符给了 `Segue`：

```
- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender{

    NSLog(@"Source Controller = %@", [segue sourceViewController]);
    NSLog(@"Destination Controller = %@", [segue destinationViewController]);
    NSLog(@"Segue Identifier = %@", [segue identifier]);

    if ([[segue identifier]
        isEqualToString:@"SimpleSegueToSecondViewController"]){

        SecondViewController *viewController = [segue destinationViewController];
        viewController.dataModel = ...
    }
}
```

#### 4.3.4. 参考 XXX

### 4.4. 在一个已有项目中添加 Storyboard

#### 4.4.1. 问题

在没有使用 Storyboard 的情况下你已经编码了一个 APP，现在你想使用 Storyboard 代替手动处理 APP 流。

#### 4.4.2. 方案

简单根据下面个步骤，你可以让 Non-Storyboard APP 使用 Storyboard：

1. 从 File 菜单，选择 New→New File...

2. 在 New File 对话框中，确保你选择了左侧 iOS 类的 Resource 子类 and 右侧的 Storyboard 选项，然后点击 Next (见图 4-24)。

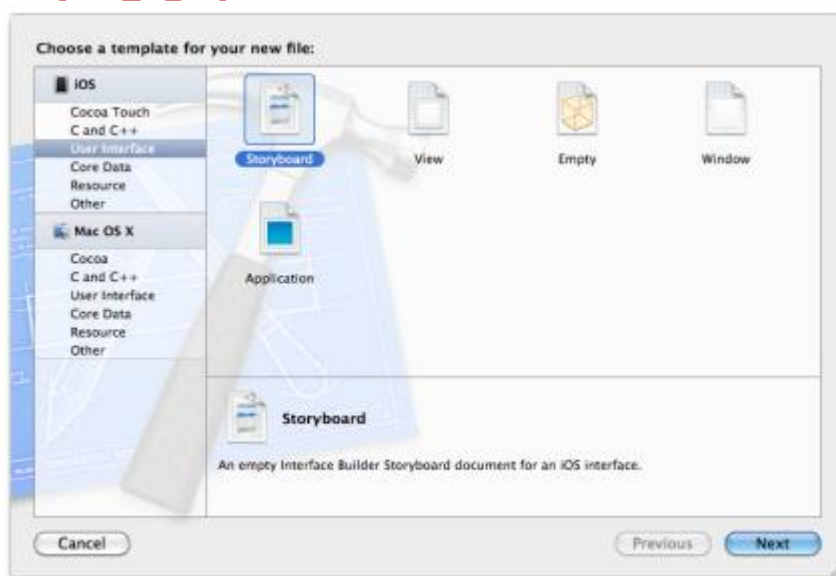


图 4-24. 将 Storyboard 增加到一个已存在 APP



3.在这个屏幕中，选择你想要创建 Storyboard 的设备序列。

如果你的 APP 是只适用于 iPhone 和 iPad 的 APP，就选择合适的设备序列。

如果你的 APP 是一个通用 APP，你现在需要选择一个设备序列，创建 Storyboard 文件，然后返回并为另一个设备序列创建 Storyboard。完成之后，点击 Next 按钮。

4. 选在选择你要保存 Storyboard 的地方。

我有一个基于 Single View Application 模板的通用 APP，所以我现在创建了 2 个 Storyboards 文件夹，分别命名为：StoryboardiPhone.Storyboard 和 StoryboardiPad.Storyboard。

5.在你的项目结构中，找出 Info.plist 文件。要注意的是.plist 可能在另一个文件后缀名保存了。例如，我把我的项目命名为 Adding a Storyboard to an Existing Project，同时我的 Info.plist 保存在 Adding a Storyboard to an Existing Project-Info.plist 下，只要你点击这个文件夹，属性列表编辑器会自动打开。

6. 如果你有这些键，把它们从.plist 文件中删除：

- NSMainNibFile (将会以 Main nib 文件基名称出现)。

- NSMainNibFile~ipad (将会以 Main nib 文件名基名称出现(iPad))。

7. 如果你有只适用于 iPhone 或者 iPad 的 APP，为 iPhone 创建一个称作 UIMainStoryboardFile 的键或者为 iPad 创建一个称作 UIMainStoryboardFile~ipad 的键。如果你有一个通用 APP，就创建 2 个键。

8.对于这些键的值，建议你创建不带.Storyboard 扩展名的 Storyboard 的文件夹名。

9. 确认保存了 .plist 文件。

10.最后一步也是最重要的一步，就是把 application:didFinishLaunchingWithOptions: 方法从 APP 委托的执行中删除。依据你的 APP 是如何建立的，这个方法可以建立不同的视图控制器和窗口等等。有了 Storyboard 后，你就不再需要它了。

#### 4.4.3. 讨论

当你在 Xcode 旧版上创建 APP 或者使用新版但是没有使用 Storyboards 时，APP 的结构将和使用了 Storyboards 的 APP 不同。这一次，基于 Storyboard 的 APP 不再使用一个主要 nib 文件作为它们的窗口。这个主要 nib 文件需要从 APP 的.plist 中删除。还需要做另一件事情，如同我们在方案部分看到的，就是要使 APP 理解在.Plist 设置之后 Storyboards 文件是什么。

所有的操作完成后，我们需要确保 APP 的委托没有混乱我们打算如何加载我们 Storyboard。每个项目不一样，你只需要确定 APP 委托没有分配任何对象到窗口的 rootViewController property。如果分配了，你的 Storyboards 将不会展示出来，那样你就要花很多个小时去找出问题的症结所在。最简单的解决办法是简单标注下整个 application:didFinishLaunchingWithOptions:方法，同时考虑在 APP 的其他地方放入你的初始化（例如，初始化和数据模型）。另一种方法是只需保留此方法，因为它不出任何可能发生变化的窗口根视图控制器对象的行的注释。

#### 4.4.4. 参考

XXX



点击这里访问: [DevDiv.com](http://DevDiv.com) 移动开发论坛