



БОРИС ОХАЙОН (BORIS OHAYON)

Разработка iOS-приложения без раскадровки (storyboard)

УЧЕБНОЕ РУКОВОДСТВО



Перевод: С. Кузнецов, 2022 г.

iOS — Start an app without a storyboard

Boris Ohayon

Sep 30, 2016 · 4 min

<https://medium.com/ios-os-x-development/ios-start-an-app-without-storyboard-5f57e3251a25>

Разработка iOS-приложения без раскадровки (storyboard)

Борис Охайон

Сен 30, 2016 · 4 мин

<https://medium.com/ios-os-x-development/ios-start-an-app-without-storyboard-5f57e3251a25>

Перевод: С. Кузнецов, 2022 г.

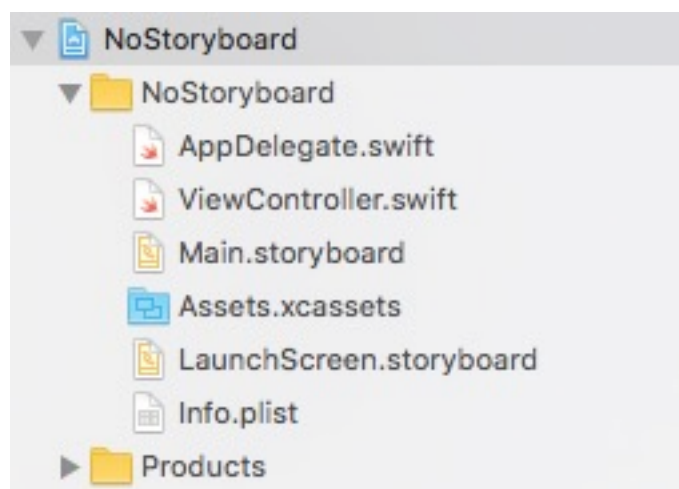




Разработка iOS-приложения без раскадровки (storyboard)

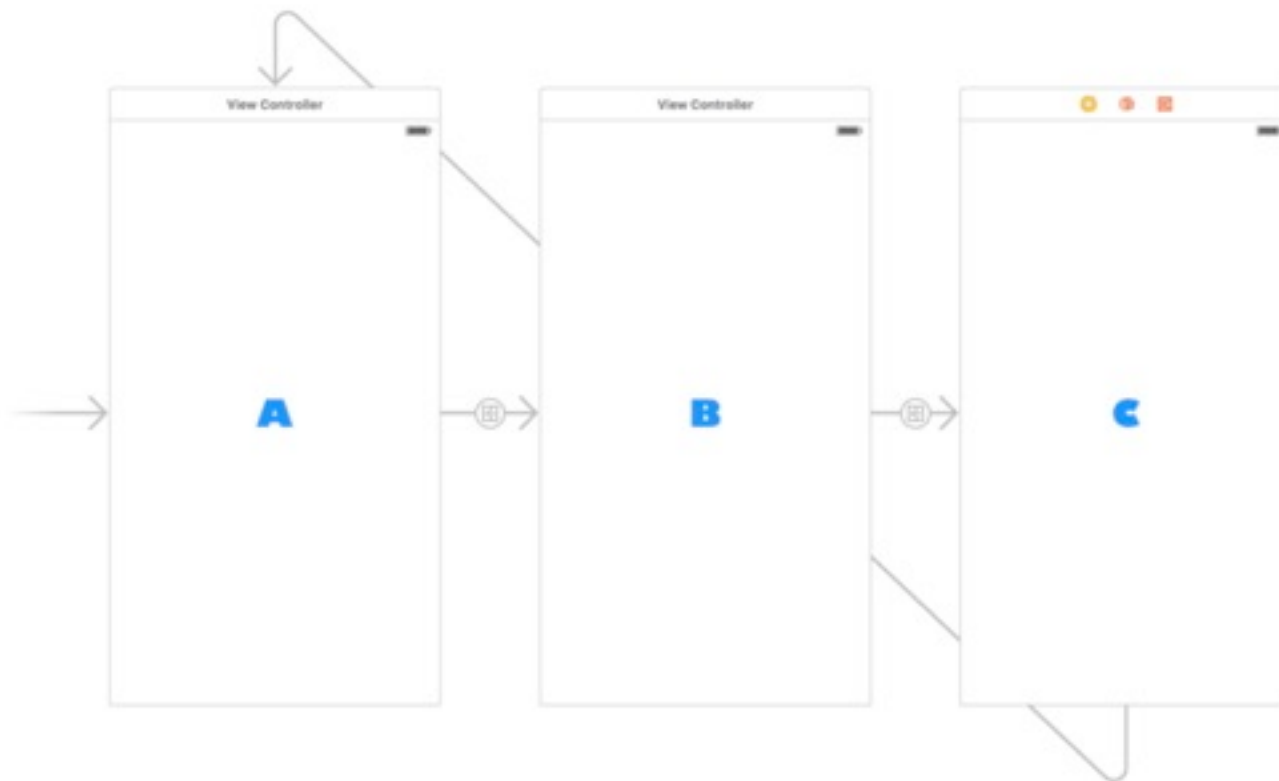
Мы изучим, как начать разработку iOS-приложения без раскадровки (storyboard)

Когда создаете приложение, то IDE-среда разработки Xcode сначала создает проект с установленной иерархией файлов:



По умолчанию, приложение создано с раскадровкой (storyboard). Раскадровки (storyboards) - хороши, они позволяют быстро помещать в нужное место визуальные элементы интерфейса и фокусироваться на некоторых других более важных вещах. Построение прототипа или макета с раскадровкой (storyboard) довольно удобно и во многих случаях является хорошим решением.





Например, мы видим здесь довольно простое приложение с тремя кнопками и при щелчке по кнопке приложение перейдет к следующему контроллеру вида. Кнопки просто поместить на место с помощью линеек в редакторе **Interface Builder (Построитель интерфейса)**, что значительно помогает визуализировать элементы интерфейса.

Когда раскадровки (storyboards) лучше не использовать?

Иногда, использование **раскадровок (storyboards)** не рекомендуется:

- Работая в команде **с управлением версиями (with version control)**
- Проектируя **сложную раскладку визуальных элементов интерфейса (complex layout)**
- Когда мы хотим легко **отслеживать ограничения (keep track of constraints)**
- Когда мы хотим **повторно использовать контроллеры вида и построить иерархию (reuse the view controllers and build a hierarchy)**

Если мы не используем **раскадровки (storyboards)**, то мы можем использовать или **NIB-файлы** или **UI-интерфейс на базе кода**. В этом учебном

примере мы рассмотрим запуск проекта с использованием только кода!
Давайте перейдем прямо к делу!

Шаг 1 — Создайте новый iOS-проект

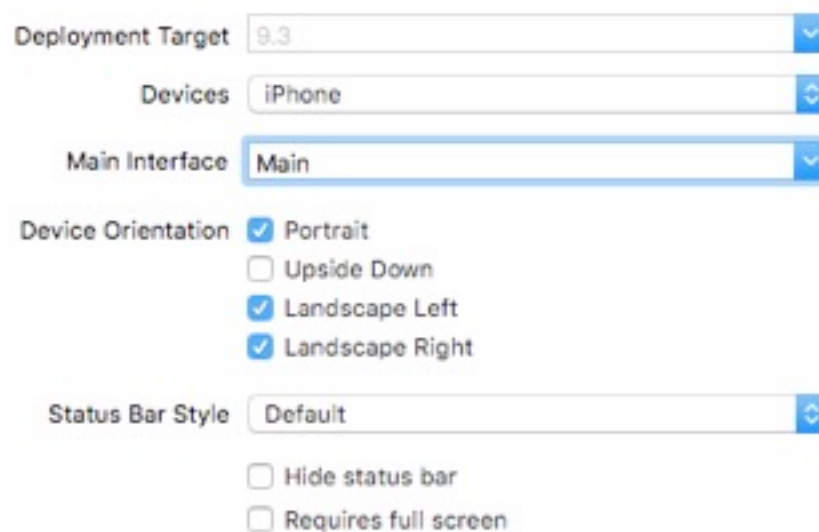
Создайте проект приложения по шаблону `Single View Application` (Приложение с одним видом), используя язык программирования `Swift`. Этот учебный пример использует язык программирования `Swift 3` и если вы будете использовать предыдущую версию языка программирования `Swift`, то незначительные отличия будут оговорены.

Шаг 2 — Удалите раскэдровку(storyboard)

На панели `Project Navigator` (Навигатор проекта) щелкните правой кнопкой мыши на `раскэдровке (storyboard)` и выберите пункт меню `delete` (удалить).

Обратите внимание на то, что удаление `раскэдровки (storyboard)` не говорит нашему приложению о не использовании `раскэдровки (storyboards)`. Еще хуже того, наше приложение будет искать `раскэдровку (storyboard)`, которую мы только что удалили, и приложение потерпит крах. Все, что мы должны сделать, должны найти место в проекте, в котором говорится об использовании `раскэдровки (storyboard)`, и изменить это место.

На панели `Project Navigator` (Навигатор проекта) щелкните по имени проекта, чтобы получить доступ к опциям. В опциях вашей цели, `General > Deployment Info` (Общее > Информация развертывания), поле `Main Interface` (Главный интерфейс) имеет значение имени `Main` (Главная). Это значение приводит к поиску `раскэдровки (storyboard)` с именем `Main` (Главная), но мы просто удалим ее.



Просто очистите значение в этом поле и наше приложение прекратит терпеть крах.

Шаг 3 — Сделайте приложение снова ЖИВЫМ

Мы еще не все сделали. Если мы сейчас запустим приложение, то получим **черный экран (black screen)**. Это вызвано тем, что мы нуждаемся в окне и в чем-то еще, что можно отобразить в этом окне.

Перейдем к делегату приложения `AppDelegate`, где мы создадим окно и первый контроллер вида `UIViewController` для отображения. В функции приложения `application` с параметрами запуска `didFinishLaunchingWithOptions`, создайте окно с заданным размером экрана:

```
func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    window = UIWindow(frame: UIScreen.main.bounds)
    return true
}
```

[исходник AppDelegate.swift](#) находится на сервисе [GitHub](#)

Обратите внимание на то, что до языка программирования `Swift 3`, синтаксис немного отличался, поскольку мы получили бы доступ к экрану с помощью `UIScreen mainScreen()`.

Теперь, когда у нас есть наше окно, добавьте любой контроллер вида `UIViewController`, который хотите, чтобы он был первым контроллером вида приложения. Здесь, мы создадим простой контроллер домашнего вида `homeViewController` класса `UIViewController`, с красным фоном, чтобы убедиться, что он есть.

```
func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    window = UIWindow(frame: UIScreen.main.bounds)

    let homeViewController = UIViewController()

    homeViewController.view.backgroundColor = UIColor.red

    return true

}
```

[view raw AppDelegate.swift](#) находится на сервисе [GitHub](#)

Обратите внимание на то, что до языка программирования `Swift 3`, синтаксис немного отличался, поскольку мы создавали бы красный цвет с помощью выражения `UIColor.redColor`.

Последний шаг должен сказать нашему окну о том, каков будет его первый контроллер вида, также известный как контроллер корневого вида `rootViewController`, являющийся свойством окна:

```
func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    window = UIWindow(frame: UIScreen.main.bounds)

    let homeViewController = UIViewController()

    homeViewController.view.backgroundColor = UIColor.red

    window!.rootViewController = homeViewController

    return true

}
```

[view raw AppDelegate.swift](#) находится на сервисе [GitHub](#)

Наконец, мы должны показать окно на экране:


```

func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    window = UIWindow(frame: UIScreen.main.bounds)

    let homeViewController = UIViewController()

    homeViewController.view.backgroundColor = UIColor.red

    window!.rootViewController = homeViewController

    window!.makeKeyAndVisible()

    return true
}

```

[исходник AppDelegate.swift](#) находится на сервисе [GitHub](#)

Шаг 4 — Запустите приложение!

Теперь, если запустите приложение, то получите хороший красный фон. Ваш проект готов к запуску без [раскадровки \(storyboard\)](#).

TL;DR

Удалите свою [раскадровку \(storyboard\)](#), очистите [Main Interface \(Главный интерфейс\)](#) в своем файле [Deployment Info \(Информации развертывания\)](#), добавьте этот код в своем делегате приложения [AppDelegate.swift](#):

```

func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    window = UIWindow(frame: UIScreen.main.bounds)

    let homeViewController = UIViewController()

    homeViewController.view.backgroundColor = UIColor.red

    window!.rootViewController = homeViewController

    window!.makeKeyAndVisible()

    return true
}

```

[view raw AppDelegate.swift](#) находится на сервисе [GitHub](#)

Вот именно так!

Теперь вы готовы начать строить приложения без раскадровки (storyboard). Первый контроллер вида (first view controller), добавленный к окну, обычно является **навигационным контроллером (navigation controller)**, позволяющим **продвигать (push)** и **представлять (present)** другие контроллеры вида на этом навигационном контроллере, но это - для другого урока!

Так что читаете? Давайте нажмем кнопку так, чтобы все могли прочитать это также!

Кроме того

Эта статья на [Medium](#)-сайте:

- Разработка iOS-приложения **без раскадровки (storyboard)** (iOS — Start an app without a storyboard)

<https://medium.com/ios-os-x-development/ios-start-an-app-without-storyboard-5f57e3251a25>