

Swift 与 Flutter 混合开发

[一滴矿泉水](#)

简介

随着跨平台技术的不断发展，越来越多的公司开始倾向跨平台语言去开发自己的应用。至于好处很多，主要是维护方便、节约成本。今天主要介绍一下在已有的 Swift 项目中通过 cocopod 集成(也是官方推荐的一种集成方式)嵌入Flutter模块。开发环境 Xcode13.2.1 、Flutter 1.26.0-18.0.pre.185 、java 1.8.0_321。

集成

- 1、创建Flutter模块，这里我们新创建个取名为 my_flutter (最好在已有的Swift项目同级目录下创建)

```
flutter create -t module my_flutter
```

- 2、通过cocopod将创建好的Flutter模块引入Swift项目(已有Swift项目名称为：Swift_Flutter)

Podfile 文件 编辑

```
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '10.0'
#忽略所有三方库警告⚠️
inhibit_all_warnings!
use_frameworks!
#此处根据自己Flutter项目实际路径填写
flutter_application_path = '../my_flutter'
#此句不可缺少
load File.join(flutter_application_path, '.ios', 'Flutter',
'podhelper.rb')
target 'Swift_Flutter' do
  install_all_flutter_pods(flutter_application_path)
end
```

添加好之后执行

```
pod install
```

```
post_install do |installer|
  flutter_post_install(installer) if defined?(flutter_post_install)
end

end
```

执行结果

```
bogon:Swift_Flutter zhanghua$ pod install
Analyzing dependencies
Downloading dependencies
Installing Flutter (1.0.0)
Installing FlutterPluginRegistrant (0.0.1)
Installing my_flutter (0.0.1)
Generating Pods project
Integrating client project
Pod installation complete! There are 3 dependencies from
the Podfile and 3 total pods installed.
bogon:Swift_Flutter zhanghua$
```

到了这里是不是就成功了呢，赶紧把项目跑起来调用一下看看吧。

调用

1、在iOS Swift代码 中调用 Flutter

我们将在应用启动的 app delegate 中创建一个 FlutterEngine，并作为属性暴露给外界。

```
import UIKit
import Flutter
import FlutterPluginRegistrant
@main
class AppDelegate: UIResponder, UIApplicationDelegate {
    var window: UIWindow?
    lazy var flutterEngine = FlutterEngine(name: "my
flutter engine")

    func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after
application launch.
        window = UIWindow(frame: UIScreen.main.bounds)
        window?.rootViewController = ViewController()
        window?.makeKeyAndVisible()

        flutterEngine.run()
```

```
GeneratedPluginRegistrant.register(with:
self.flutterEngine)    Registrant 登记者

    return true
}
}
```

控制器中调用

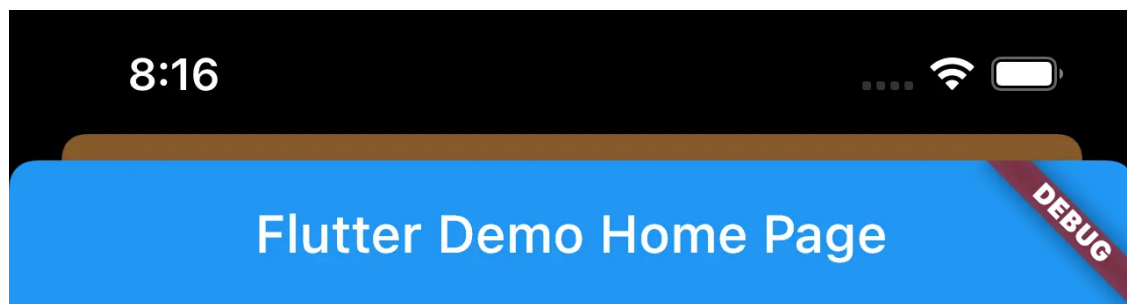
```
import UIKit
import Flutter

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        view.backgroundColor = UIColor.brown
        // Do any additional setup after loading the view.
    }

    override func touchesBegan(_ touches: Set<UITouch>,
with event: UIEvent?) {
        //搜索到了很多中获取控制器跳转的方法，感觉这种获取 控制器跳转
        最为流畅 （自我感觉）
        let flutterEngine = (UIApplication.shared.delegate
as! AppDelegate).flutterEngine
        let flutterViewController =
FlutterViewController(engine: flutterEngine, nibName: nil,
bundle: nil)
        present(flutterViewController, animated: true,
completion: nil)
    }
}
```

- 启动项目点击屏幕



You have pushed the button this many times:

0

