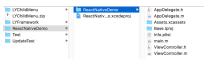
IOS群：335930567（收水勿扰）

这里介绍了，如何使用上面提到的工具：

- Xcodegen
- ReactNative这个程序的程序，及其所需配置

1、首先我创建一个ios项目，如果你已经有项目可以自己分析，直接跳到对项目进行操作



2、由于本人是新ios一开发的，因此只在一个项目的基础上进行配置，而又只用项目来了，创建一个空白项，用于在这上次形成所需要文件，为了方便管理，我把文件夹统统放在目录

3、接下来，打开终端

```
cd 项目目录
```

```
touch package.json
```

```
open package.json
```

react-native后面的version的版本用严格固定，其中版本号来可在官网查询可以
dependencies后面的key的react-native的版本因其对可能的兼容要求
也可以使用npm info react 和 npm info react-native来查看当前的最新版本

```
{
  "name": "ReactNativeDemo",
  "version": "0.0.1",
  "private": true,
  "scripts":
    "start": "node node_modules/react-native/local-cli/cli.js start"
  },
  "dependencies": {
    "react": "15.6.1",
    "react-native": "0.39.2"
  }
}
```

```
npm install
```

4、使用cocoapods配置ReactNative程序

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'ReactNativeDemo' do
  # Comment the next line if you're using Swift or would like to use dynamic
  # frameworks
  # use_frameworks!

  # Pods for ReactNativeDemo
  pod 'React', :path => '../node_modules/react-native', :subspecs => [
    'Core',
    'RCTImage',
    'RCTNetwork',
    'RCTWebSocket', # 这个模块是用于调试功能的
    # 在这里继续添加所需的模块
  ]
end
```

```
pod init
```

```
# Uncomment the next line to define a global platform for your project
target 'ReactNativeDemo' do

  # Comment the next line if you're using Swift or would like to use dynamic
  # 在使用混编的项目情况下，请在这里取消注释后进行下面的 :path
  pod 'React', :path => '../node_modules/react-native', :subspecs => [
    'Core',
    'RCTImage',
    'RCTNetwork',
    'RCTWebSocket', # 这个模块是用于调试功能的
    # 在这里继续添加所需的模块
  ]

end
```

```
pod install
```

5、创建index文件

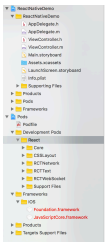
```
cd 项目目录
```

```
touch index.ios.js
```

把index文件夹，编辑如的代码

```
'use strict';

import React from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
} from 'react-native';

class HelloGlobeview extends React.Component {
  render() {
    let contents = this.props["arrows"].map(
      arrow => (<Text key={arrow.name}>{arrow.name}:{arrow.value}{"\n"}</Text>
      )
    );
    return (
      <View style={styles.container}>
        <Text style={styles.welcomeTitle}>
          列表数据展示
        </Text>
        <Text>
        <Text style={styles.arrows}>
          {contents}
        </Text>
        </Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  welcomeTitle: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  },
  arrows: {
    textAlign: 'center',
    color: '#333333',
    marginBottom: 5,
  },
});

// 注册一个模块的名称
AppRegistry.registerComponent('ReactNativeDemo', () => HelloGlobeview);    // 这与配置文件的
个名字，第一个名字是项目的名字，第二个名字是上面写的名字
```

6、接下来就进行开项目，编写一部分

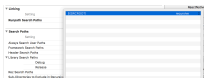


编辑如代码：

```
#import "AppDelegate.h"
#import "RCTRootView.h"
#import "RCTBundleURLProvider.h"
@interface AppDelegate ()
@end
```

对应如图，修改对应部分

修改完，需要如下操作：



```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
    (NSDictionary *)launchOptions
{
  // Poplate AppGlobals
  // You can add your own globals here
  // Define URL where to find the JavaScript bundle
  NSURL *jsCodeLocation = [NSURL
                           URLWithString:@"http://localhost:8081/
                                           index.ios.bundle?platform=ios"];

  RCTRootView *rootView =
  [[RCTRootView alloc] initWithBundleURL: [jsCodeLocation
                                           moduleName:
                                           @"ReactNativeDemo"
                                           initialProperties:
                                           @{
                                             @"arrows": @[
                                               @{
                                                 @"name": @"Alice",
                                                 @"value": @"42"
                                                 },
                                               @{
                                                 @"name": @"Bob",
                                                 @"value": @"18"
                                                 },
                                               ],
                                             }
                                           launchOptions: nil
                                           ];
  UIViewController *vc = [UIViewController alloc] init];
  rootView.frame = [UIScreen mainScreen].bounds;
  [vc.view addSubview:rootView];
  self.window = [UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds
                 rootViewController: vc];
  [self.window makeKeyAndVisible];
  return YES;
}
```

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
{
  // Poplate AppGlobals
  // You can add your own globals here
  // Define URL where to find the JavaScript bundle
  NSURL *jsCodeLocation = [NSURL
                           URLWithString:@"http://localhost:8081/index.ios.bundle?
platform=ios"];

  RCTRootView *rootView =
  [[RCTRootView alloc] initWithBundleURL: [jsCodeLocation
                                           moduleName:
                                           @"ReactNativeDemo"
                                           initialProperties:
                                           @{
                                             @"arrows": @[
                                               @{
                                                 @"name": @"Alice",
                                                 @"value": @"42"
                                                 },
                                               @{
                                                 @"name": @"Bob",
                                                 @"value": @"18"
                                                 },
                                               ],
                                             }
                                           launchOptions: nil
                                           ];
  UIViewController *vc = [UIViewController alloc] init];
  rootView.frame = [UIScreen mainScreen].bounds;
  [vc.view addSubview:rootView];
  [vc.view setBackgroundColor:[UIColor whiteColor]];
  self.window = [UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
  self.window.rootViewController = vc;
  [self.window makeKeyAndVisible];
  return YES;
}
```

8、到这一步还是有错误，你需要开启服务器

```
cd 项目目录
```

```
npm start
```

然后就可以从开项目，或者直接可以：

```
react-native run-ios
```

下面的是自定义，直接点击，如果你的项目名称是ios-demo模式，会报错的原因，你需要用xcode模式

报错提示：

```
dyld:Library not loaded:/usr/local/lib/libglog.0.dylib
  Referenced from:
  /Library/Developer/CoreSimulator/Profiles/Runtimes/iOS.simruntime/Contents/Resources/RuntimeRoot/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation
  Reason: image not found
```