REACT-NATIVE FOR IOS

基础篇

by 许恒

提纲

1.准备

2.特点

3.安装

4.布局

5.手势

6.组件

7.API

8.自定义API

9.调试

10.RN-OC的通信机制

11.发布

12.集成到现有App

13.总结

14.需要解决的难题

15.参考资料

准备

- Reactjs
- Es6
- 还有得有一台mac

特点

- learn once, write anywhere
- Native Components
- Asynchronous Execution
- Touch Handling
- Flexbox and Styling
- Polyfills
- Extensibility

安装

- 安装 nodejs 4.0 && watchman

```
$ npm install -g react-native-cli
$ react-native init AwesomeProject
```

- 演示

布局

- 宽度单位和像素密度
- flex布局
- 图片布局
- 绝对定位和相对定位
- padding、margin
- 文本

手势(一)

- Touchable*
  - TouchableHighlight
  - TouchableOpacity
  - TouchableWithoutFeedback
- 事件
  - onPress
  - onPressIn
  - onPressOut
  - onLongPress
- 演示

手势(二)

- View的手势事件(从上到下依次触发)
  - onStartShouldSetResponder
  - onMoveShouldSetResponder
  - onResponderGrant
  - onResponderMove
  - onResponderRelease
- 演示

手势(三)

- 事件对象
  - changedTouches、identifier、locationX、locationY、pageX、pageY、target、timestamp、touches
- View发生嵌套时,基于冒泡机制触发事件
- 演示

组件

- View
- Text
- Image
- ListView
- ActivityIndicatorIOS
- TabBarIos

API

- AlertIOS
- Animated
- AppRegistry
- AsyncStorage
- ProgressViewIOS
- StatusBarIOS
- PushNotificationIOS

自定义API

```objc
#import <Foundation/Foundation.h>
#import "RCTBridgeModule.h"

@interface Mop : NSObject <RCTBridgeModule>

@end
```

```objc
#import "MOP.h"

@implementation Mop

RCT_EXPORT_MODULE();

RCT_EXPORT_METHOD(hello: (NSString*) name callback: (RCTResponseSenderBlock) callback) {
  NSLog(@"hello %@",name);
   [NSThread sleepForTimeInterval:3.0f];
  callback(@[@"我已经收到你的来信"]);
}

@end
```

自定义API(二)

```
var React = require('react-native')
var Mop = require('react-native').NativeModules.Mop

var {
    StyleSheet,
    Text,
    View,
    TouchableHighlight
} = React;

var PageOne = React.createClass({
    componentDidMount(){
        console.log(require('react-native').NativeModules)
    },
    render: function(){
        return (
                <TouchableHighlight style={styles.outside} onPress={this.pressHandle}>
                    <View style={styles.container}>
                        <Text style={styles.one}>1</Text>
                    </View>
                </TouchableHighlight>
            )
    },
    pressHandle(){
        Mop.hello('XXX',(msg)=>{
            console.log(msg)
        })
    }
```

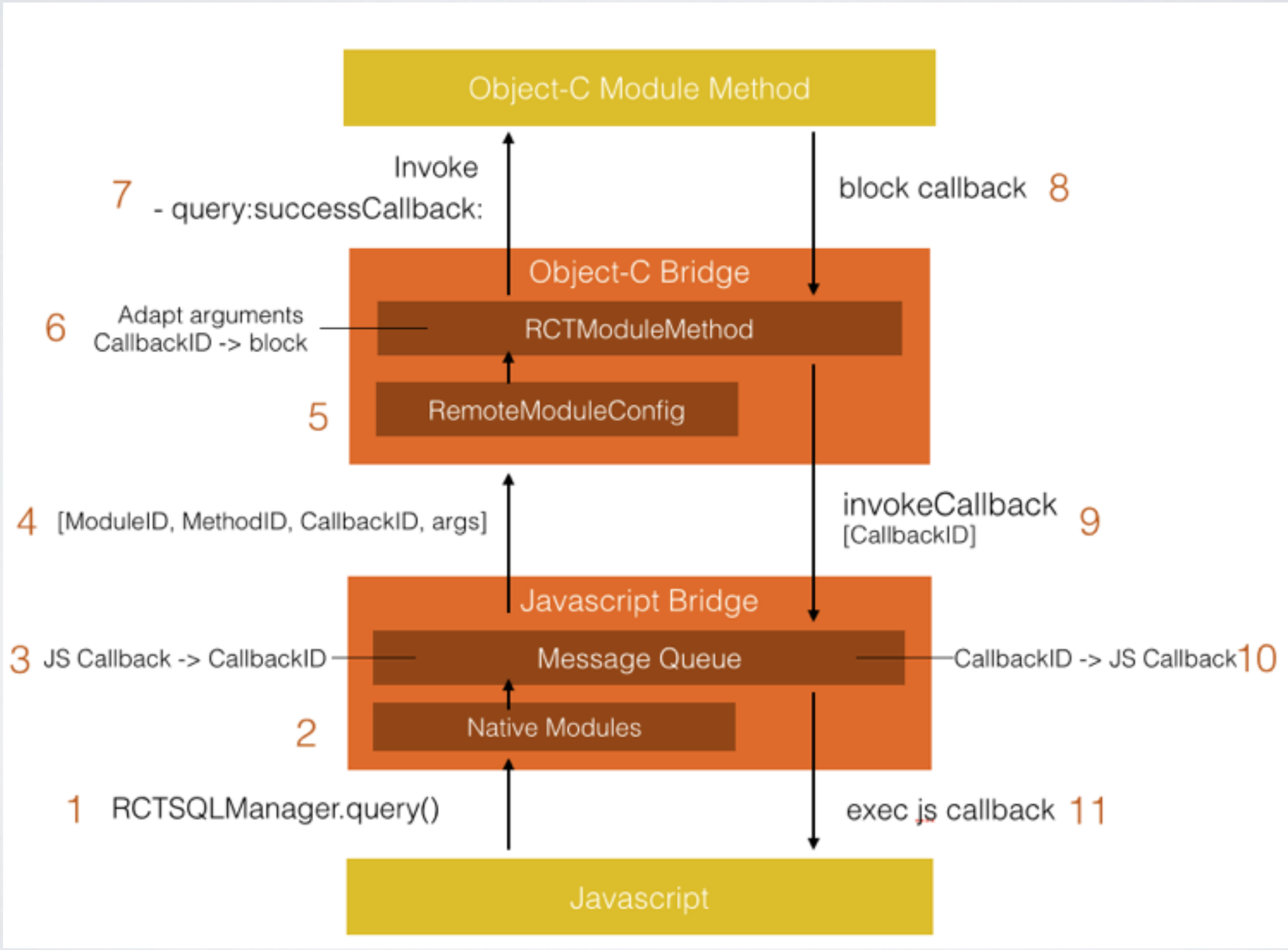
调试

- pc chrome debug
- mobile debug
- 演示

RN-OC通信机制

- 普通的JS-OC通信
  - 在iOS上，Native通过-[UIWebView stringByEvaluatingJavaScriptFromString:]调用Web,而Web则是通过设置WebView iframe的src,搭建JSBridge进行通讯
- RN-OC通信
  - 基于iOS自带的JavaScriptCore作为JS的解析引擎，以此建立通信机制

RN-OC通信机制


Object-C Module Method
Invoke
7
- query:successCallback:
block callback 8
Object-C Bridge
6
Adapt arguments
CallbackID -> block
RCTModuleMethod
5
RemoteModuleConfig
4 [ModuleID, MethodID, CallbackID, args]
invokeCallback
[CallbackID]
9
Javascript Bridge
3 JS Callback -> CallbackID
Message Queue
CallbackID -> JS Callback 10
2
Native Modules
1 RCTSQLManager.query()
exec js callback 11
Javascript

发布(一)

发布到appstore时,需要将js和oc的代码一起打包

实时更新的功能就不存在了?

发布(二)

- http://microsoft.github.io/code-push/index.html

- 通过每次向服务器拉取最新的jsbundle来更新app

发布(三)

- 服务端(提供2个在线的接口)

    1. 返回最新的jsbundle版本号
    2. 返回最新的jsbundle文件

- OC端

    3. 根据版本号判断是否拉取最新的jsbundle文件
    4. 如果有新版本，就将最新的jsbundle文件拉取到本地，存为本地资源
    5. 以后每次运行app只用读本地资源

集成到现有APP

1. 安装 cocoapods && nodejs
2. 用cocoapods安装react-native，在根目录新增Podfile文件，pod install
3. 新增index.ios.js入口文件, 并添加内容
4. 添加 ContainerView 到App
5. 添加 RCTRootView 到 ContainerView
6. 启动开发服务器
7. 编译、启动

查看详细地址

总结(一)

- 优点
  - 更新代码不依赖发版，但需要“jsbundle”代码版本管理器
  - 基于jsx、css-layout的布局，对于前端人员更加友好
  - 所有的操作都是异步执行，流畅度和响应性都非常好
  - chrome、手机调试so easy
  - 复用reactjs,减少学习成本
- 缺点
  - 学习成本较高相较于之前的hybrid方案
  - iOS和android only的组件越来越多，多端复用只有底层核心代码
  - bug和坑还是比较多

总结(二)

使用场景

替换之前hybrid那套方案

需要解决的问题

- jsbundle版本管理器
- 自定义Component
- 自定义API

参考资料

- https://facebook.github.io/react-native/
- https://github.com/facebook/react-native
- https://github.com/ele828/react-native-guide
- http://gold.xitu.io/entry/55fa202960b284975l9db23f
- http://microsoft.github.io/code-push/index.html#getting_started
- https://github.com/Microsoft/react-native-code-push
- http://guides.cocoapods.org/using/getting-started.html

谢谢