



中国移动开发者大会
Mobile Developer Conference China 2016

58同城React Native开发实践

- iOS视角

58无线技术部 彭飞

- ✓支持的系统版本
- ✓组件库支持的业务功能
- ✓ListView性能

iOS RN 调研

2015.6

- ✓RN集成进已有项目
- ✓载体页与统一导航
- ✓JS如何打包给Native调用
- ✓Native与JS的事件交互
- ✓组件的封装粒度

双平台RN基础调研

2016.3

- ✓热更新包大小及内置资源大小的控制
- ✓Diff存储及更新方案
- ✓异常回滚机制
- ✓基于二进制的diff 还是基于文本的diff

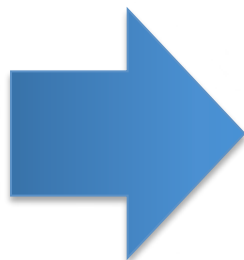
热更新调研

2016.4



您今天还可免费发帖 2 条，本月还可免费发帖 3 条。

小 区	1-30字	>
区 域	请选择区域	>
户 型	室 厅 卫	
面 积	平方米	
租 金	元/月	
标 题	8-28字	
描 述	房源配置等 至少10字	

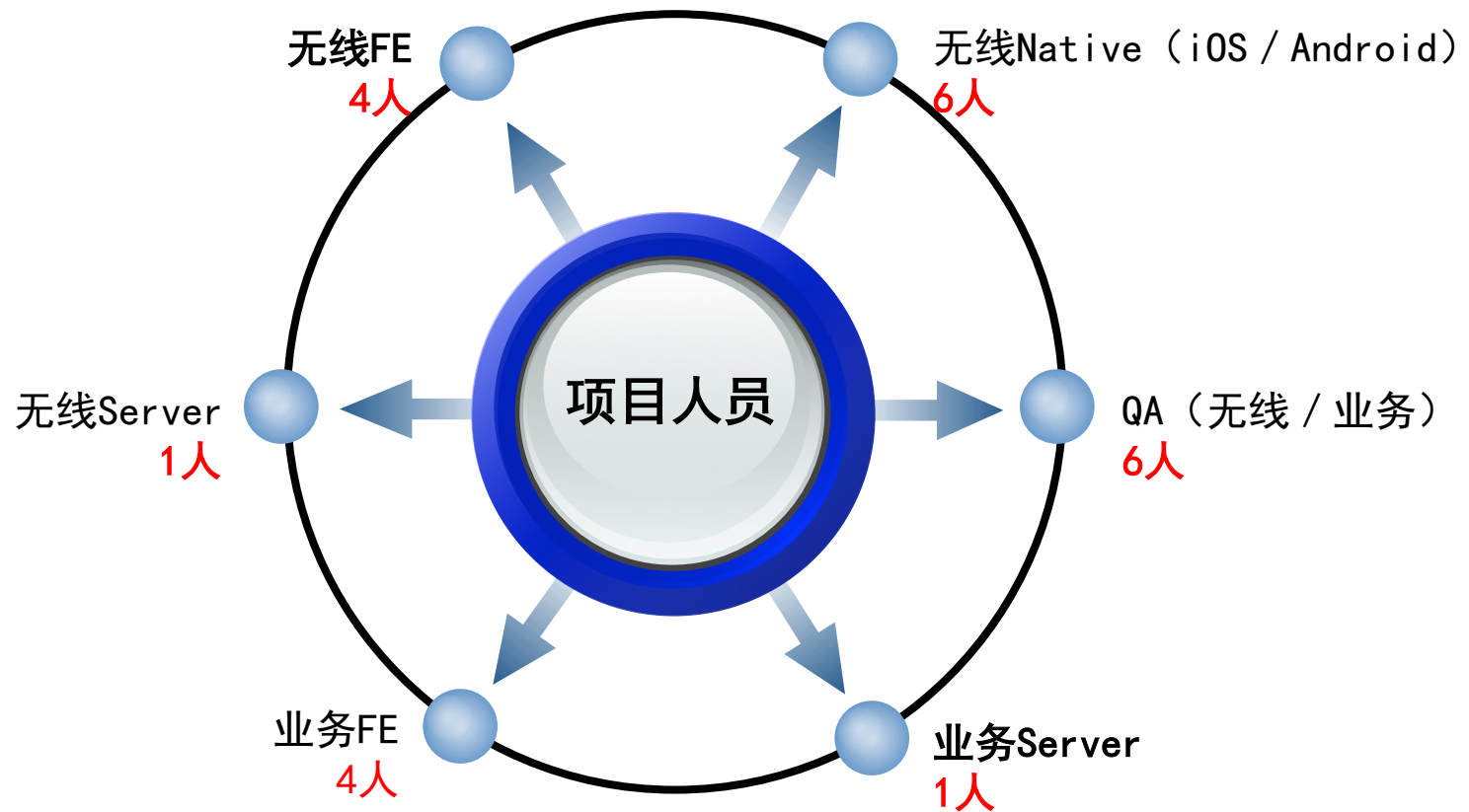


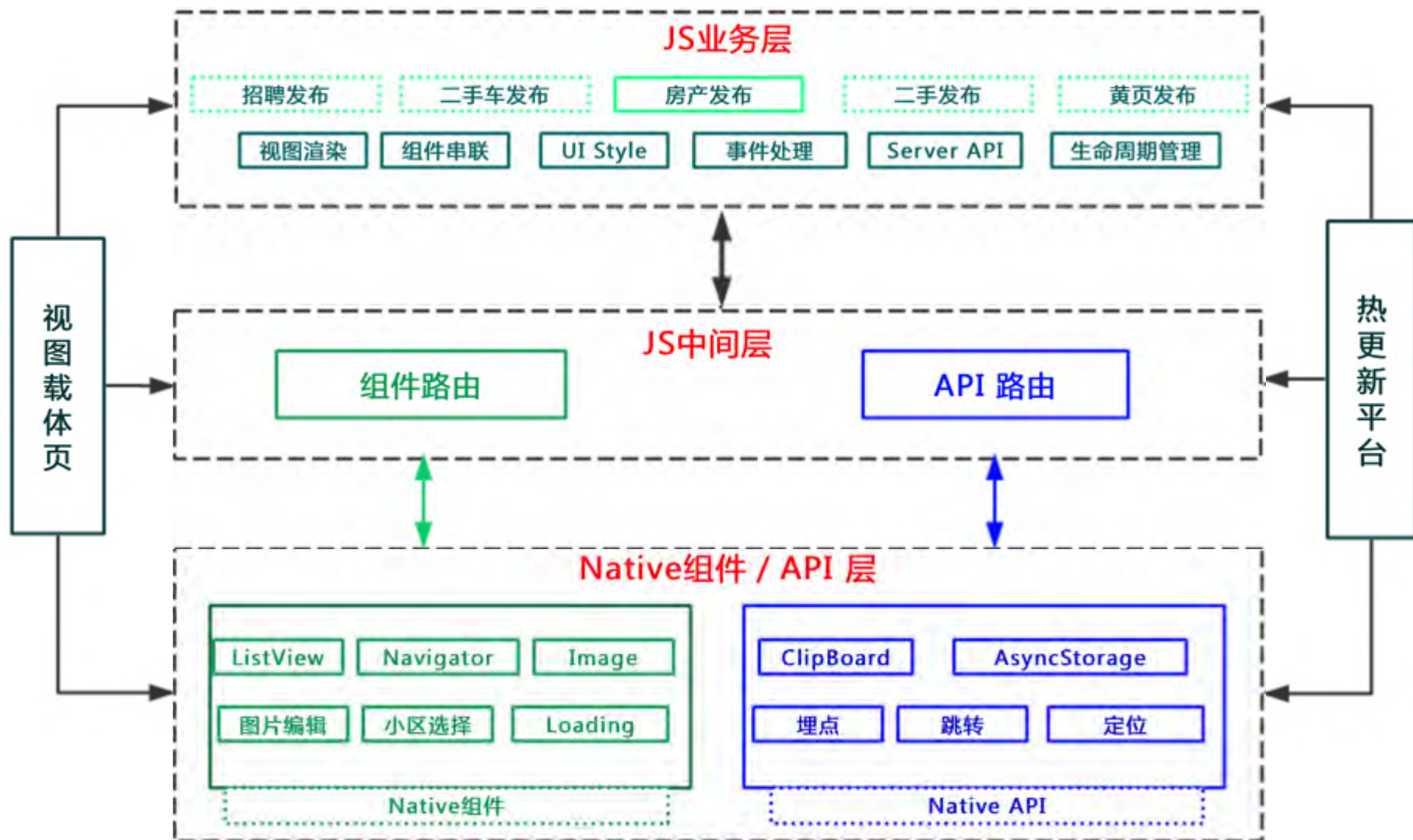
您今天还可免费发帖 2 条，本月还可免费发帖 3 条。

添加图片

想让帖子更有吸引力？快来上传照片吧~

小区	1-25字，不能填写电话	>
区域	请选择区域	>
户型	请选择户型	>
面积	请输入房源面积	平米
租金	请输入租金	元/月
标题	标题为8-30字，不能填写联系方式	
描述	请填写房源描述，至少10个字，可填写对租客的要求，周边环境及商业配套等等	






```
s.subspec 'Core' do |ss|
  ss.source_files = "WBReactNativeLibrary/react-native/React/**/*.*"
  ss.exclude_files = "**/__tests__/*", "IntegrationTests/*"
  ss.frameworks = "JavaScriptCore"
  ss.libraries = "stdc++"
end
```

#载体页

```
s.subspec 'WBRCTCarrier' do |ss|
  ss.dependency 'WBReactNativeLibrary/Core'
  ss.source_files = "WBReactNativeLibrary/WBRCTCarrier/*.{h,m}"
  ss.preserve_paths = "WBReactNativeLibrary/WBRCTCarrier/*.js"
end
```

#下面是自定义封装的组件

```
s.subspec 'WBRCTComponent' do |ss|
  ss.dependency 'WBReactNativeLibrary/Core'
  ss.source_files = "WBReactNativeLibrary/WBRCTComponent/*.{h,m}"
  ss.resources = "WBReactNativeLibrary/WBRCTComponent/WBRCTComponentResource/*.{bu
```

- ▼ WBReactNativeLibrary
 - ▶ ART
 - ▶ Core
 - ▶ RCTActionSheet
 - ▶ RCTAdSupport
 - ▶ RCTCameraRoll
 - ▶ RCTGeolocation
 - ▶ RCTImage
 - ▶ RCTLinkingIOS
 - ▶ RCTNetwork
 - ▶ RCTPushNotification
 - ▶ RCTSettings
 - ▶ RCTText
 - ▶ RCTVibration
 - ▶ RCTWebSocket
 - ▶ Support Files
 - ▶ WBRCTBuildinResource
 - ▶ WBRCTCarrier
 - ▶ WBRCTComponent

组件名称	描述	组件名称	描述
WBArea	区域商圈选择	WBLocation	定位
WBCustomDialog	自定义弹窗	WBMultiSelector	多选
WBErroView	错误页面	WBPageDispatcher	跳转
WBFinishPage	页面关闭	WBPay	支付
WBHSAreaSelect	小区选择	WBPhotoSelect	图片编辑
WBHttpHeader	请求头	WBToast	Toast提示
WBInitialParams	初始化参数	WBUserLog	埋点
WBLoadingBar	加载bar	WBVoiceRecorView	语音
WBLoadingView	加载view	WBShare	分享

中国移动 17:48 100%

整套出租

您今天还可免费发帖 2 条，本月还可免费发帖 3 条。


添加图片
想让帖子更有吸引力？快来上传照片吧~

小区 1-25字，不能填写电话

区域 请选择区域

户型 请选择户型

面积 请输入房源面积 平米

租金 请输入租金 元/月

标题 标题为8-30字，不能填写联系方式

请填写房源描述，至少10个字，可填写对租客的要求，周边环境及商业配套等等



中国移动 17:48 100%

整套出租

我的位置：北京朝阳大山子

雁塔

碑林

莲湖

新城

未央

长安

灞桥

临潼

阎良

高新区

沣渭新区

曲江新区

杨凌

西安周边

 空间 ▾ 人员 创建 ...

- 标题栏带图片的Dialog
- LoadingBar
- Toast
- 加载中Loading
- 小区选择
- 区域商圈选择
- 行业选择(7.2.0添加)
- 单选选择器
- 多选无联动选择器
- 埋点
- 定位
- 语音
- 异常页面
- 登陆界面调起
- 触发RN更新
- 通用数据存储
- 分享
- 发布入口

页面 / ... / 组件接口协议

区域商圈选择

创建：彭飞，最新修改于：2016 05 23

调起区域选择组件的接口：

WBAreaManager:showArea(pageTitle,catelId, resultCallback)

#pageTitle: 进入区域选择界面后导航bar上的title

#catelId: 当前发布页面所属的类别id

#resultCallback, 选择区域返回值callback,

i) 若用户未选择，不执行回调。

ii) 若用户按照正常流程选择，需返回一个数组，数组里面的每一个元素都是一个字典，按照选择顺序排列。

字典里面包含3个元数据：paramname, text, value

返回示例：

[{"paramname":"localArea","text":"宣武","value":"1141"}, {"paramname":"localDiduan","text":"椿树街道","value":"6856"}, ...]

 赞 成为第一个赞同者

 编写评论...

无标签

```
18 RCT_EXPORT_MODULE();
19
20 /**
21  返回示例:
22  [{"paramname":"localArea","text":"宣武","value":"1141"},
23   {"paramname":"localDiduan","text":"椿树街道","value":"6856"}, ...]
24  */
25 RCT_EXPORT_METHOD(showArea:(NSString*) pageTitle
26                   cateId:(NSString*) cateId
27                   resultCallback:(RCTResponseSenderBlock)callback)
28 {
29     //必填参数校验
30     if (cateId == nil || cateId.length <= 0) {
31         return;
32     }
33
34     dispatch_async(dispatch_get_main_queue(), ^{
35         //保存回调block
36         self.rnResultCallBack = callback;
37
38         self.title = pageTitle;
39         self.strCateId = cateId;
40         self.arrSelectModel = [NSMutableArray array];
41         self.arrSubWidget = [NSMutableArray array];
42
43         //在UIWindow上调出界面
44         [self showAreaOnScreen:pageTitle];
45     });
46 }
```

```
26 //小区选择的使用方法
27 ▼ press:function(val){
28
29     //埋点
30     var mycars=new Array("Saab","Volvo","BMW")
31     NativeModules.WBUserLog.statistics('main','sysmsg',mycars);
32
33     //小区选择 (null表示没有传递附近小区的数据)
34     var showCommunityArea = NativeModules.WBAreaManager;
35     showCommunityArea.showArea("整租房",'100',(dataArray)=>{
36         console.log('选择小区后返回的数据'+dataArray);
37     ▲ });
38 ▲ },
39
```



```
52 /**
53  * Provides the interface needed to register a bridge module.
54  */
55 @protocol RCTBridgeModule <NSObject>
56
57 /**
58  * Place this macro in your class implementation to automatically register
59  * your module with the bridge when it loads. The optional js_name argument
60  * will be used as the JS module name. If omitted, the JS module name will
61  * match the Objective-C class name.
62  */
63 #define RCT_EXPORT_MODULE(js_name) \
64 RCT_EXTERN void RCTRegisterModule(Class); \
65 + (NSString *)moduleName { return @"#js_name; } \
66 + (void)load { RCTRegisterModule(self); }
67
```

默认设置:

`RCT_EXPORT_MODULE();`

指定别名:

`RCT_EXPORT_MODULE(js_name);`

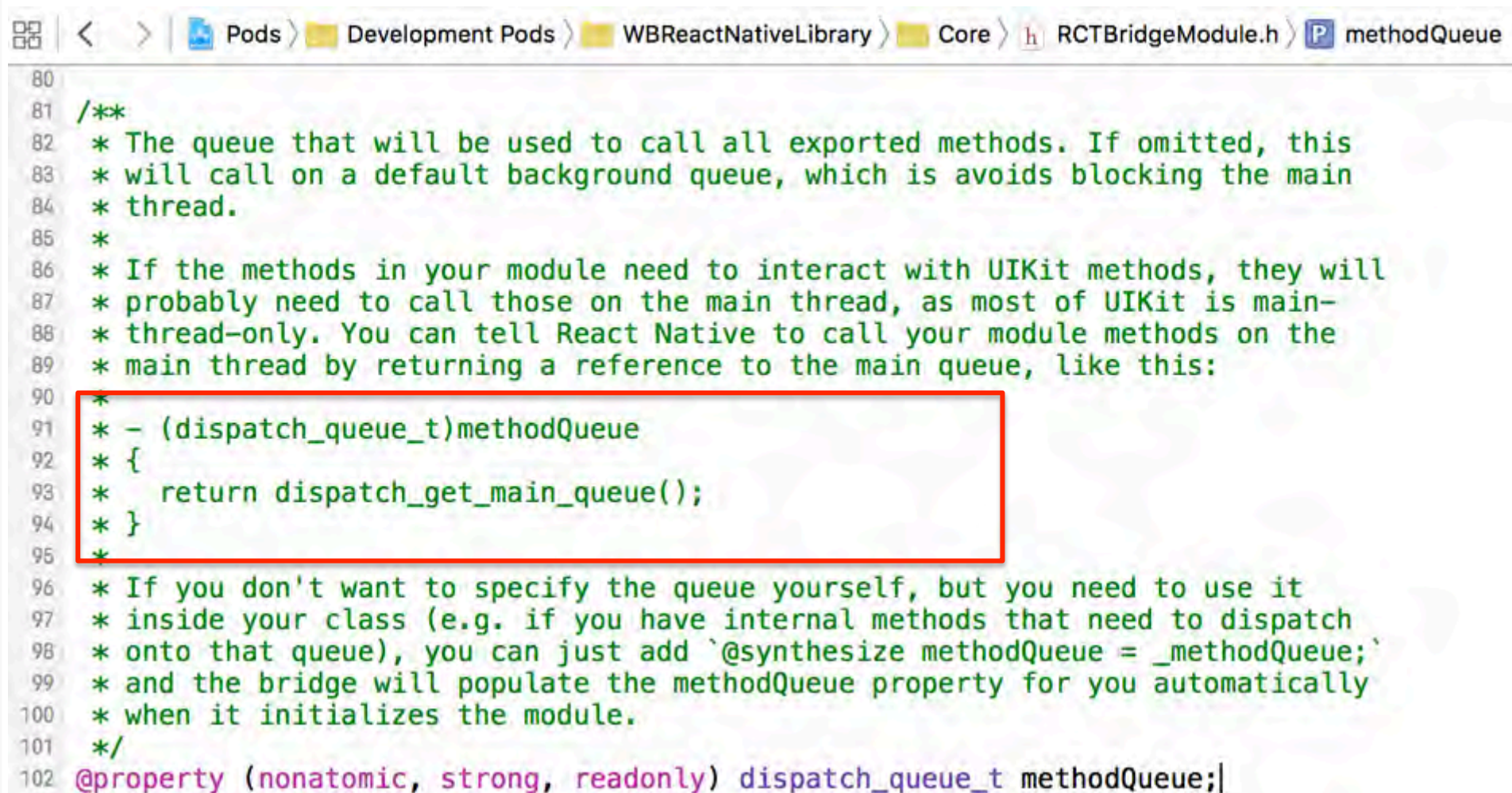
```
144 #define RCT_EXPORT_METHOD(method) \  
145     RCT_REMAP_METHOD(, method)|  
146  
147 /**  
148  * Similar to RCT_EXPORT_METHOD but lets you set the JS name of the exported  
149  * method. Example usage:  
150  *  
151  * RCT_REMAP_METHOD(executeQueryWithParameters,  
152  *   executeQuery:(NSString *)query parameters:(NSDictionary *)parameters)  
153  * { ... }  
154  */  
155 #define RCT_REMAP_METHOD(js_name, method) \  
156     RCT_EXTERN_REMAP_METHOD(js_name, method) \  
157     - (void)method  
158
```

默认设置:

```
RCT_EXPORT_METHOD(show:(NSString*)message  
duration:(NSInteger)duration) {...}
```

指定别名:

```
RCT_REMAP_METHOD(jsShowName, show:  
(NSString*)message duration:(NSInteger)duration)  
{...}
```

```
80
81 /**
82  * The queue that will be used to call all exported methods. If omitted, this
83  * will call on a default background queue, which is avoids blocking the main
84  * thread.
85  *
86  * If the methods in your module need to interact with UIKit methods, they will
87  * probably need to call those on the main thread, as most of UIKit is main-
88  * thread-only. You can tell React Native to call your module methods on the
89  * main thread by returning a reference to the main queue, like this:
90  *
91  * - (dispatch_queue_t)methodQueue
92  * {
93  *     return dispatch_get_main_queue();
94  * }
95  *
96  * If you don't want to specify the queue yourself, but you need to use it
97  * inside your class (e.g. if you have internal methods that need to dispatch
98  * onto that queue), you can just add `@synthesize methodQueue = _methodQueue;`
99  * and the bridge will populate the methodQueue property for you automatically
100  * when it initializes the module.
101  */
102 @property (nonatomic, strong, readonly) dispatch_queue_t methodQueue;
```

```
NSArray *imageList = @[@"http://foo.com/bar1.png",  
                        @"http://foo.com/bar2.png"];  
  
NSDictionary *props = @{@"images" : imageList};  
  
RCTRootView *rootView = [[RCTRootView alloc] initWithBridge:bridge  
                        moduleName:@"ImageBrowserApp"  
                        initialProperties:props];
```



```
@interface RCTWBInitialParams : NSObject<RCTBridge  
  
//初始化跳转协议数据  
+(void) configJumperData:(NSDictionary*) jumperDi  
  
//根据数据类型获取对应的数据  
+(NSMutableDictionary*) produceInitialData:(int) dataType;  
@end
```





**解耦合
提高开发效率**

iOS / Android平台差异处理

接口参数校验，特殊事件处理

自定义组件 / API封装


```
--  
82 // 线上环境报错不调起 Native 的 API  
83 if(error === 'error'){  
84     return error  
85 }  
86  
87 let {title="", content="", negativeButtonText="取消", positiveButtonText="确定", imageUrl} = param;  
88  
89 // 如果是不同回调, 先取消上次订阅, 再重新注册。  
90 if (callback !== lastCallback) {  
91     subscription && subscription.remove();  
92     subscription = DeviceEventEmitter.addListener('RCTDidButtonSelectedEvent', callback);  
93 }  
94  
95 if (Platform.OS === 'ios') {  
96  
97     // ios 默认图片地址为 shield  
98     RCTImageDialog.show(imageUrl="shield", title, content, negativeButtonText, positiveButtonText);  
99  
100 } else {  
101     param.imageUrl = "";  
102     RCTImageDialog.show(param);  
103  
104 }
```


步骤一

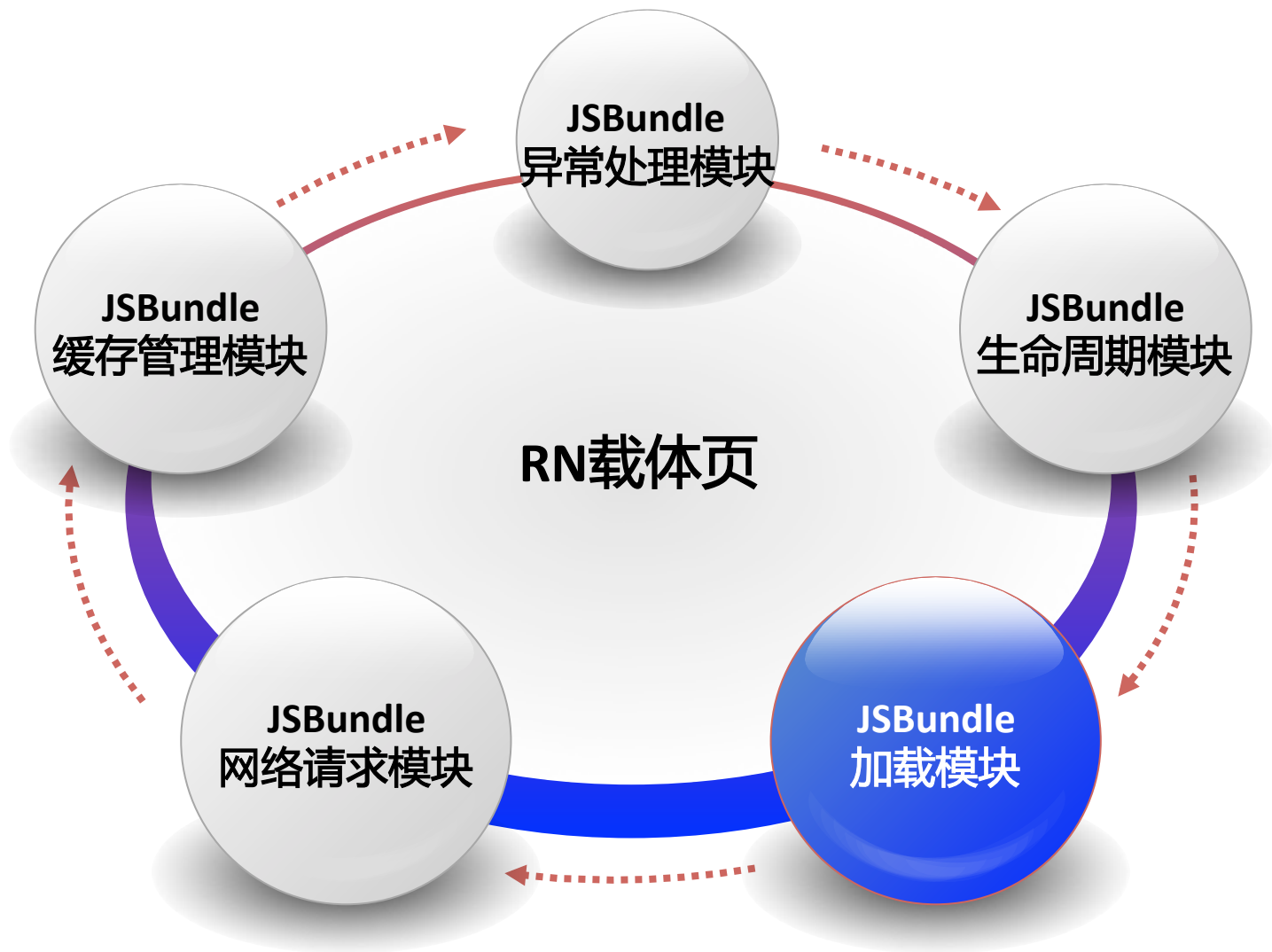
- 申请git权限
- 下载中间层代码pod

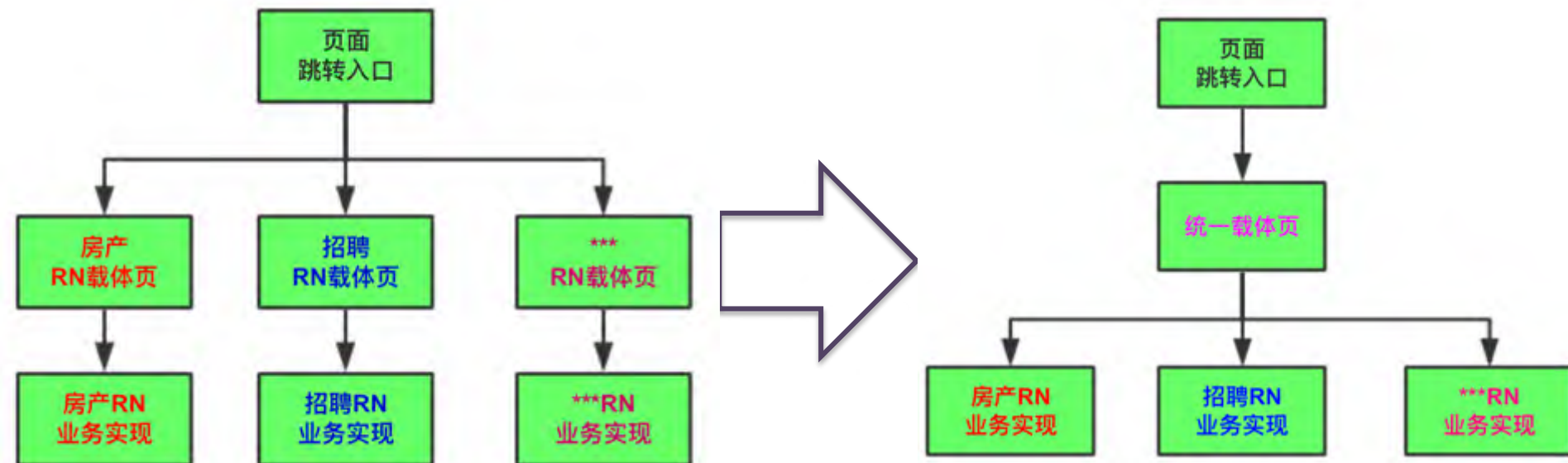
步骤二

- 查看confluence使用文档
- 项目中引用

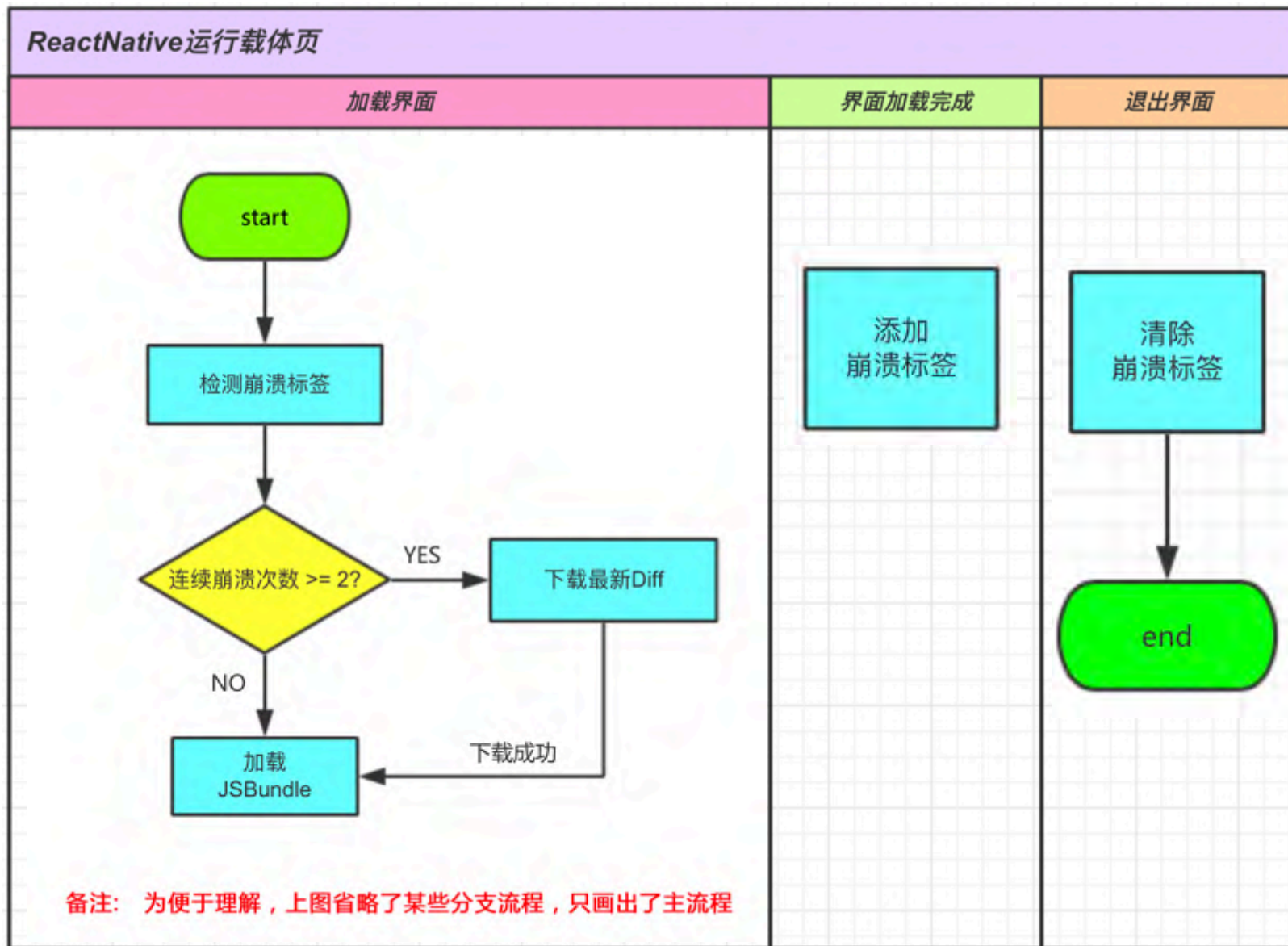
步骤三

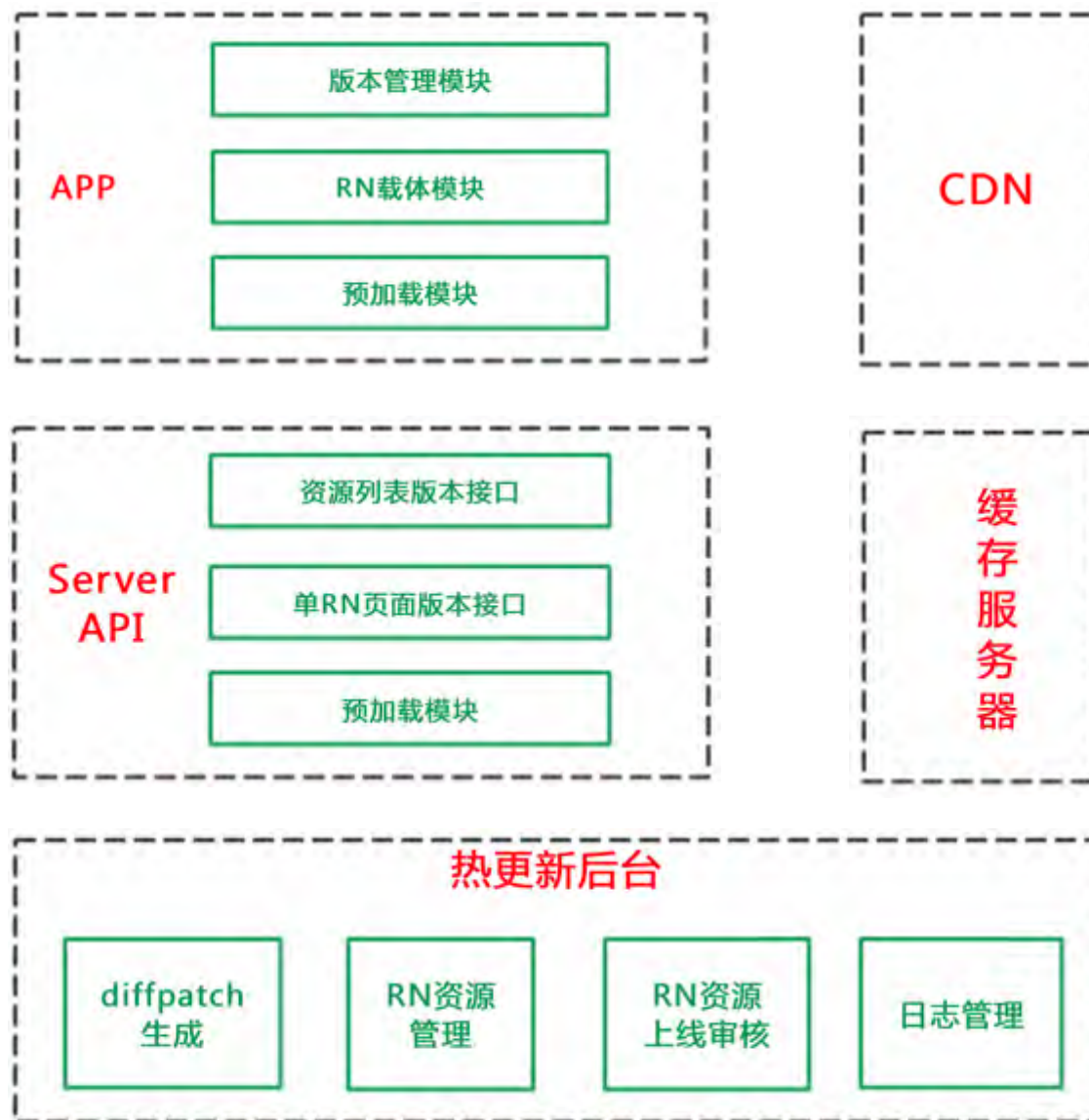
- 安装含RN的58 APP
- 进入载体页调试

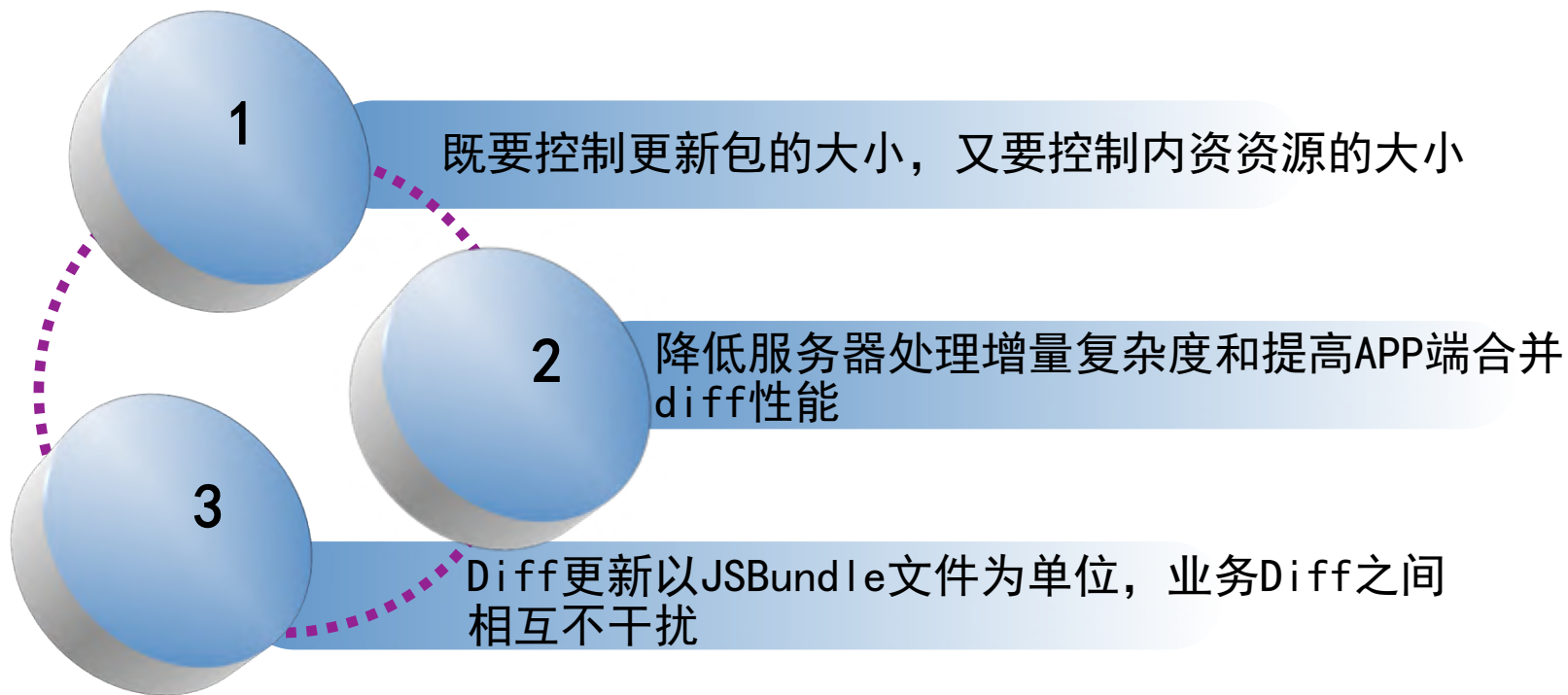


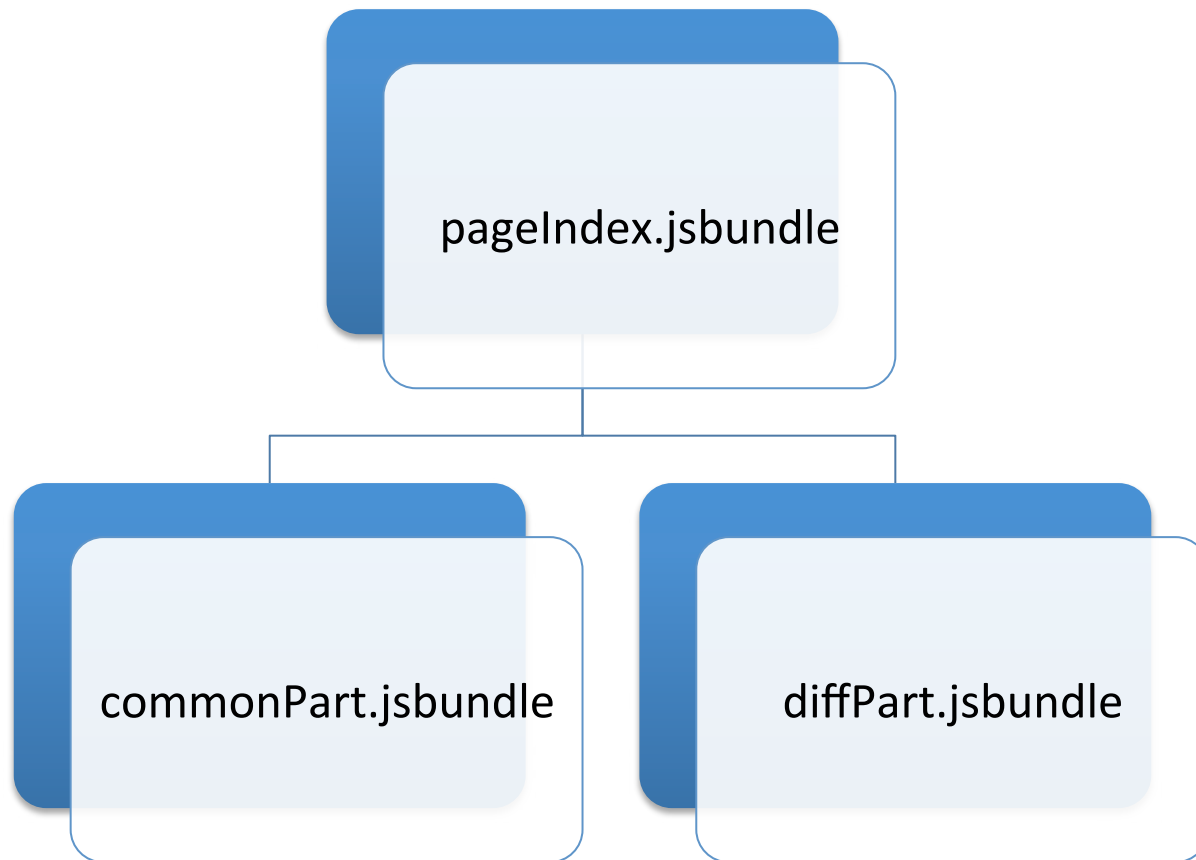


```
Pods > Development Pods > WBReactNativeLibrary > Core > RCTAssert.m > RCTFatal()
119
120 void RCTFatal(NSError *error)
121 {
122     _RCTLogNativeInternal(RCTLogLevelFatal, NULL, 0, @"%@", error.localizedDescription);
123
124     RCTFatalHandler fatalHandler = RCTGetFatalHandler();
125     if (fatalHandler) {
126         fatalHandler(error);
127     } else {
128         #if DEBUG
129             @try {
130                 #endif
131                 NSString *name = [NSString stringWithFormat:@"%@: %@", RCTFatalExceptionName, error.localizedDescription];
132                 NSString *message = RCTFormatError(error.localizedDescription, error.userInfo[RCTJSStackTraceKey], 75);
133                 [NSException raise:name format:@"%@ ", message];
134                 #if DEBUG
135             } @catch (NSException *e) {}
136             #endif
137         }
138     }
139
140     void RCTSetFatalHandler(RCTFatalHandler fatalhandler)
141     {
142         RCTCurrentFatalHandler = fatalhandler;
143     }
144
145     RCTFatalHandler RCTGetFatalHandler(void)
146     {
147         return RCTCurrentFatalHandler;
148     }
```









步骤一

- 新建空白js文件

步骤二

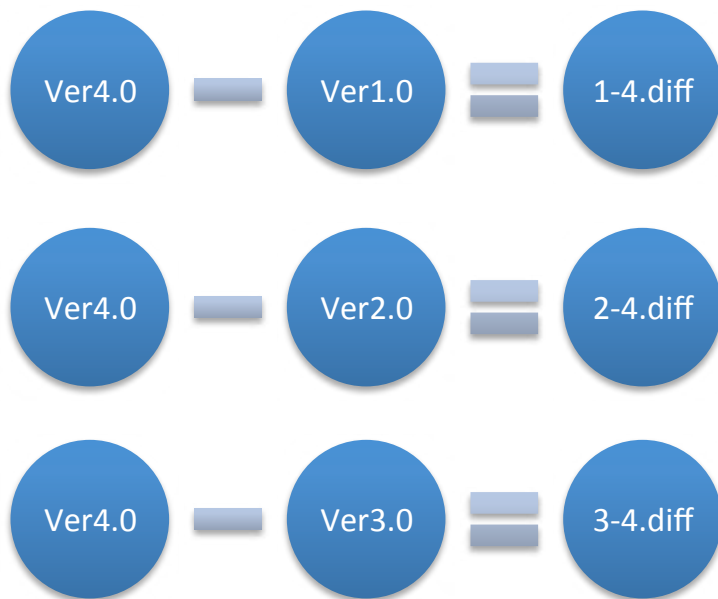
- 仅import react及react-native

步骤三

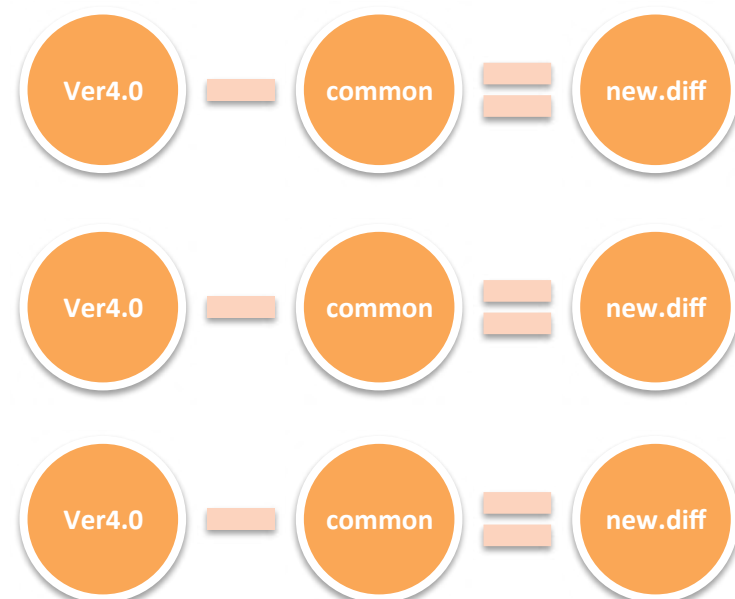
- 通过curl命令将blank.ios.js文件编译成common.ios.jsbundle

通过google-diff-match-patch算法
比较公共jsbundle与入口页面jsbundle，生成差异jsbundle

WBRNHotUpdate				
名称 ^ 修改日期 大小 种类				
▼ 2	今天 下午4:30	--	文件夹	
2_old.patch	2016年8月9日 下午12:08	260 KB	修补程序文件	
2.patch	2016年8月12日 上午11:32	263 KB	修补程序文件	
BundleException.plist	今天 下午4:30	315 字节	Property List	
common.bundle	2016年7月28日 上午11:07	531 KB	捆绑包	
JSBundleIndex.plist	2016年8月12日 上午11:32	305 字节	Property List	

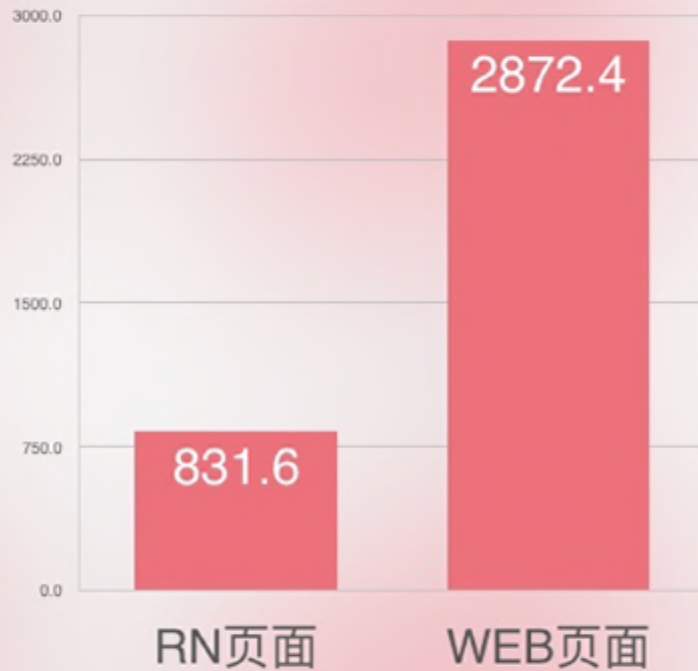


基于版本的更新

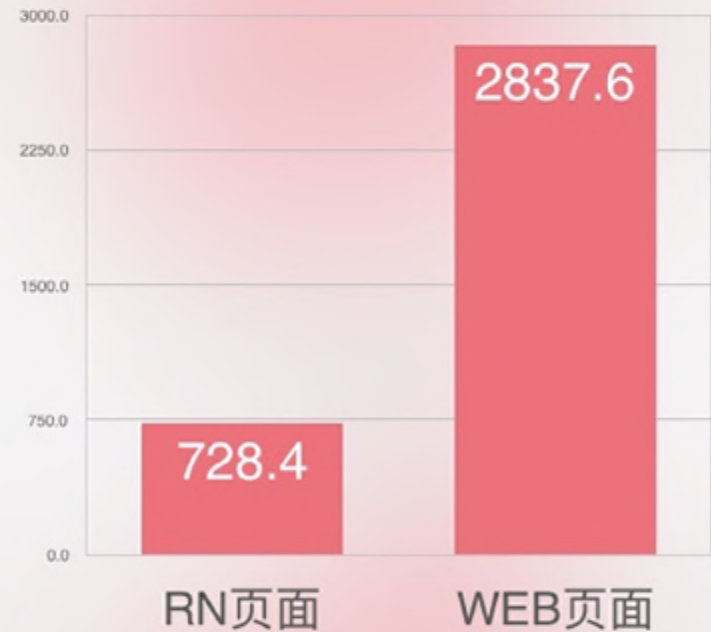


基于Common的更新

Android



iOS



Q&A

谢谢！



58无线技术公众号，更多干货在这里



写代码累了，下载58转转，
闲置物品赚点酒钱