

58 RN 页面 开发 流程

Lemon Ray



RN 其主要原理是由一套代码编译之后，可运行在双平台上

```
'use strict';

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

class HelloWorldApp extends Component {
  render() {
    return (
      <View>
        <Text>Hello world!</Text>
        <Text>This is 58 RN world!</Text>
      </View>
    );
  }
}

AppRegistry.registerComponent('Muba', () => HelloWorldApp);
```

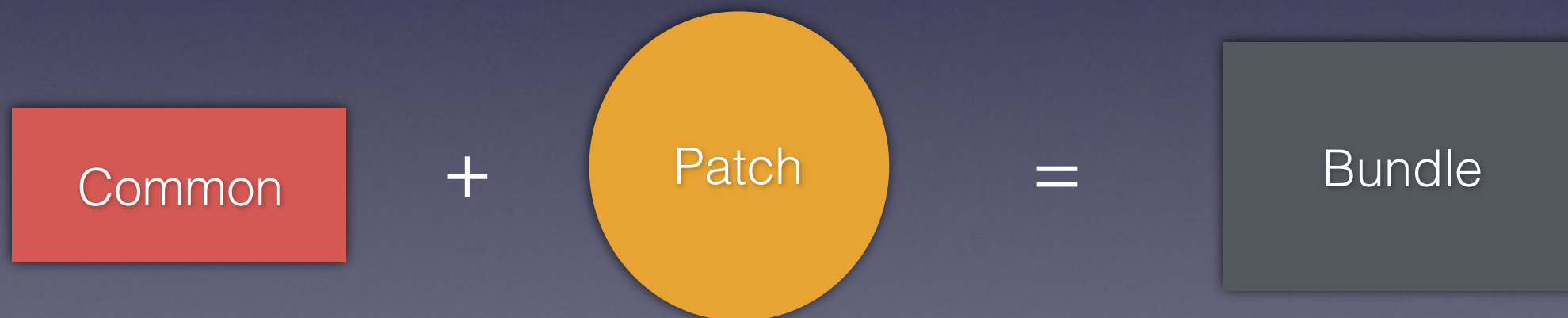
IOS可用的
BUNDLE

ANDROID可用的
BUNDLE



简单理解一下热更新的方案

```
index.android.bundle x
81870
81871   TopicGroup;
81872 }, "Wuba/App/components/TopicGroup.js");
81873 _d(682 /* Wuba/App/lib/CompareVersion.js */, function(global, require, module, exports) {Object.defineProperty(
81874
81875
81876
81877
81878   function CompareVersion(higer,lower){var sep=arguments.length>2&&arguments[2]!==undefined?arguments[2]:',';
81879   var higherAry=higer.split(sep),
81880       lowerAry=lower.split(sep);
81881   var l=Math.max(higherAry.length,lowerAry.length);
81882   var zeros=''.repeat(l);
81883
81884   for(var i=0;i<l;i++){
81885
81886     var high=parseInt(higherAry[i]||'');
81887     var low=parseInt(lowerAry[i]||'');
81888
81889     if(high>low){
81890       return 1;
81891     }
81892
81893     if(high<low){
81894       return -1;
81895     }
81896   }
81897   return 0;
81898 }exports.__esModule=true;
81899
81900 //module.exports={compareVersion};
81901 //require("Wuba/App/lib/CompareVersion.js");
81902 //require(185);
81903 ;(function(){var e=/^utf-8:b6790176c70b5dc92bc8c5875708872e7cfb09f3+/
```



类比

Web	RN
1.一个正常的APP	1.IOS特殊包， 安卓任意测试包
2.正常开发 浏览器调试	2.正常开发 APP调试
3.App.js	3.中间层
4.ftp上线	4.热更新平台
5.配置跳转协议	5.配置跳转协议

1.所需的包

流程

开发

测试

上线

Web

RN

代理

bundle

部署对应地址的代码

Common + Patch的热
更新流程

web页面

开发	任意 58 APP
测试—上线	任意 58 APP

rn页面

开发(只需要运行bundle)	IOS 特殊的测试包 Android 任意测试包
测试 — 上线(热更新流程)	正常的线上包或测试包

注1：本次分享的内容全是针对开发的，直到第四部分上线

注2：目前Native测试包支持的都是react-native 0.28 版的注意，一定是0.28版的，不然，你跑不通的.....

2.正常开发以及调试

web开发

写代码

代理

部署对应地址的代码

RN开发

我们不需要热更新流程，只需要编译生成bundle，使用APP跑通bundle即可



bundle

再次强调：目前Native测试包支持的都是react-native 0.28 版的注意，一定是0.28版的，不然，你跑不通的.....

git地址: <http://gitlab.58corp.com/leitao/share-rn-demo>

Name	Last Update	Last Commit > 9010bed2 – some change on app.js
📁 App	16 minutes ago	👤 leitao some change on app.js
📁 middle	24 minutes ago	👤 leitao 真完善
📄 .gitignore	24 minutes ago	👤 leitao 真完善
📄 README.md	22 minutes ago	👤 leitao readme.md
📄 index.android.js	about an hour ago	👤 leitao 初始化
📄 index.ios.js	about an hour ago	👤 leitao 初始化
📄 package.json	24 minutes ago	👤 leitao 真完善

README.md

说明

关于58RN 前端必须知道的N件事的demo，供大家后面测试用 项目中包括最新的测试包，以及一个小demo

IOS测试地址(取inhouse包就可以)


```
http://build.58corp.com/job/i58client_WBCube_7.6.0_rn_loading_update/ws/package/Products/
```

安卓测试包地址

```
http://build.58corp.com/job/58WuxianClient_release_7.6.0_modify_rn_loading/ws/v7.6.0_58585858_20161217_15.21__58WuxianClie
```


package.json

真完善

 leitao authored 2 months ago

d459cd0f

Browse Code »

 package.json 236 Bytes

EditRawBlameHistoryPermalinkRemove

```
1  {
2    "name": "HelloWorldApp",
3    "version": "0.0.1",
4    "private": true,
5    "scripts": {
6      "start": "node node_modules/react-native/local-cli/cli.js start"
7    },
8    "dependencies": {
9      "react": "15.2.0",
10     "react-native": "^0.28"
11   }
12 }
```

index.ios.js

```
/**
 * IOS主入口
 */
'use strict';
import App from './App/app';
```

index.android.js

```
/**
 * Android主入口
 */
'use strict';
import App from './App/app';
```

```
'use strict';

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

class HelloWorldApp extends Component {
  render() {
    return (
      <View>
        <Text>Hello world!</Text>
        <Text>This is 58 RN world!</Text>
      </View>
    );
  }
}

AppRegistry.registerComponent('Wuba', () => HelloWorldApp);
```

第一种：编译指令

```
react-native bundle --entry-file ./index.ios.js --bundle-output index.ios.bundle --  
bundle-encoding "utf-8" --platform "ios"
```

第二种：开启服务

```
react-native start
```

本地开启服务，请求bundle

开启服务也是提供编译的过程，不过请求地址是

IOS

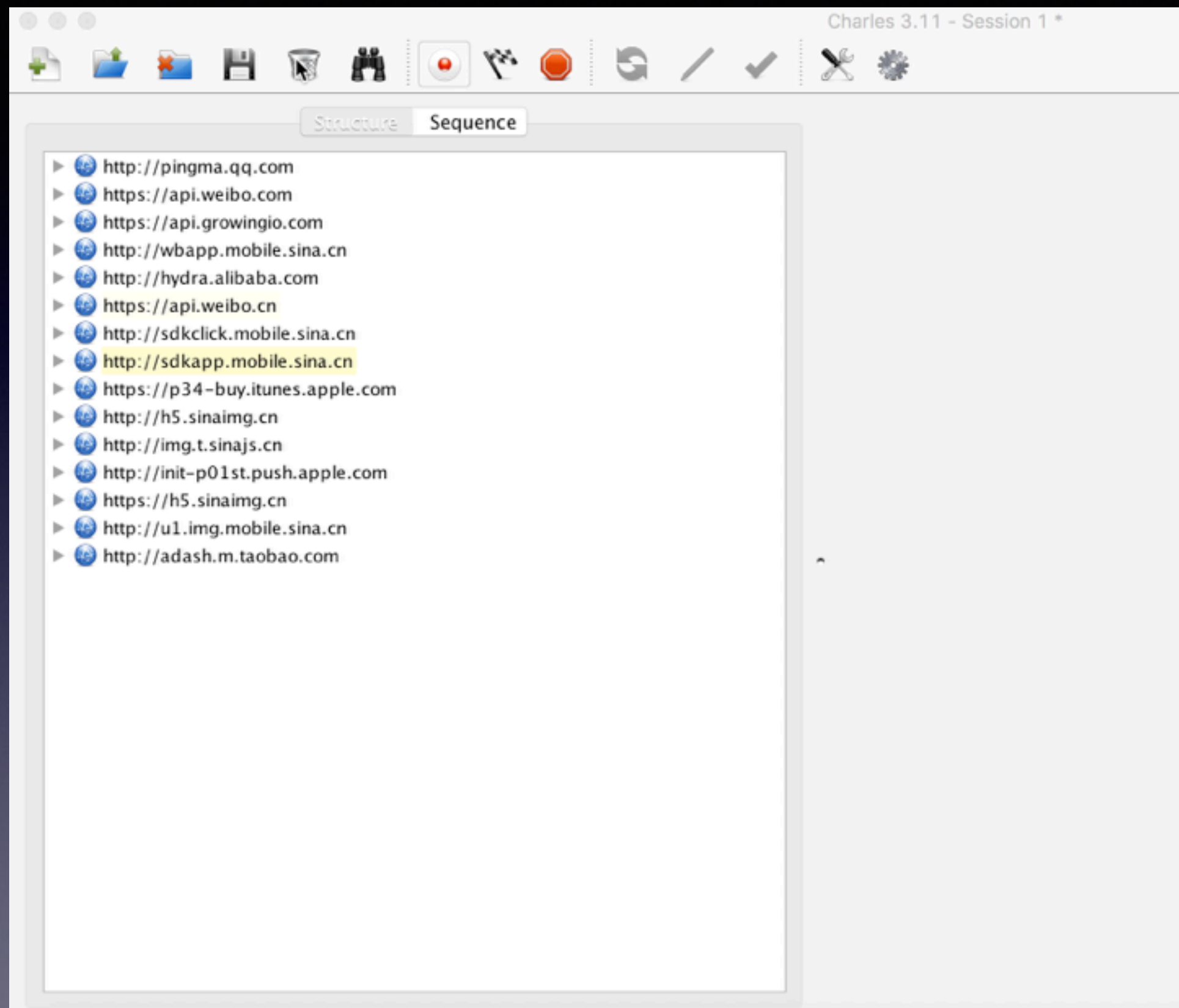
<http://localhost:8081/index.ios.bundle?platform=ios&dev=true&hot=false&minify=false>

Android

<http://localhost:8081/index.android.bundle?platform=android&dev=true&hot=false&minify=false>

share-rn-demo — -bash — 80x24

```
LemondeMacBook-Pro:share-rn-demo lemonray$ react-native bundle --entry-file ./index.ios.js --bundle-output index.ios.bundle --bundle-encoding "utf-8" --platform "ios"
```



demo展示IOS







15:56



Could not connect to development server.

Try the following to fix the issue:

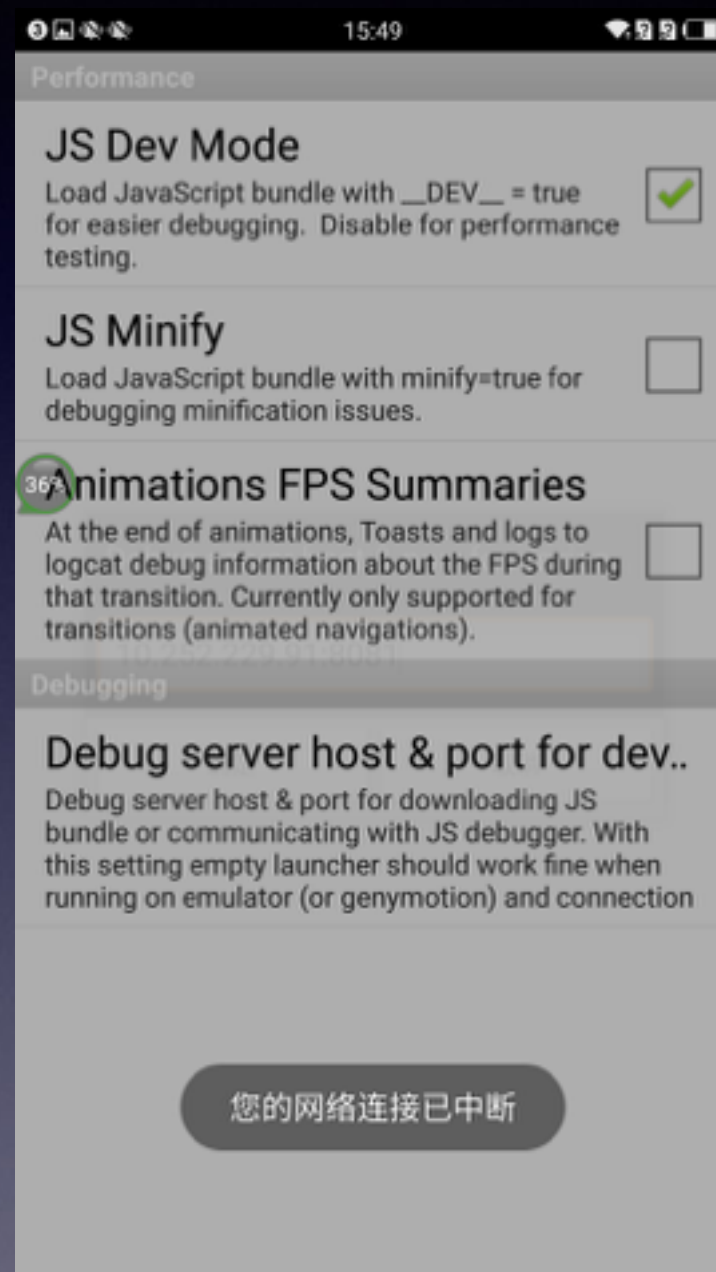
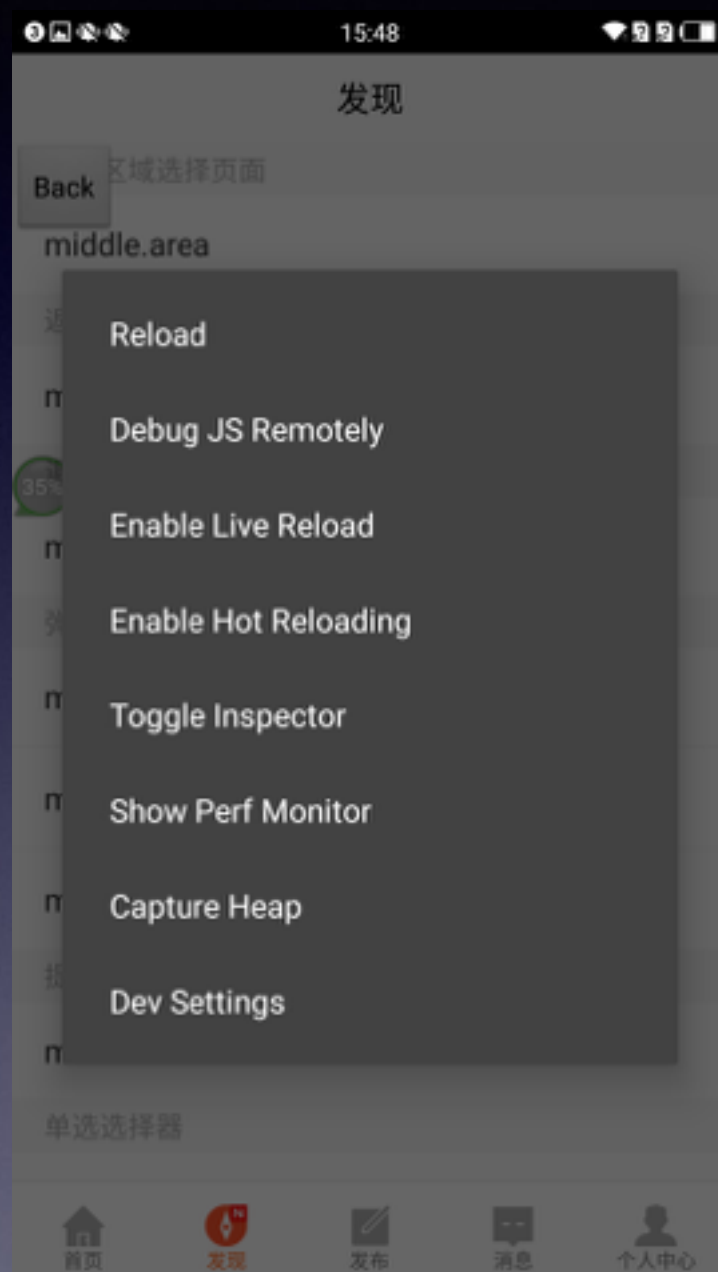
- Ensure that the packager server is running
- Ensure that your device/emulator is connected to your machine and has USB debugging enabled - run 'adb devices' to see a list of connected devices
- If you're on a physical device connected to the same machine, run 'adb reverse tcp:8081 tcp:8081' to forward requests from your device
- If your device is on the same Wi-Fi network, set 'Debug server host & port for device' in 'Dev settings' to your machine's IP address and the port of the local dev server - e.g. 10.0.1.1:8081

URL: <http://localhost:8081/index.android.bundle?platform=android&dev=true&hot=false&minify=false>

onFailure
DevServerHelper.java:196

execute
RealCall.java:140

RELOAD



app.js

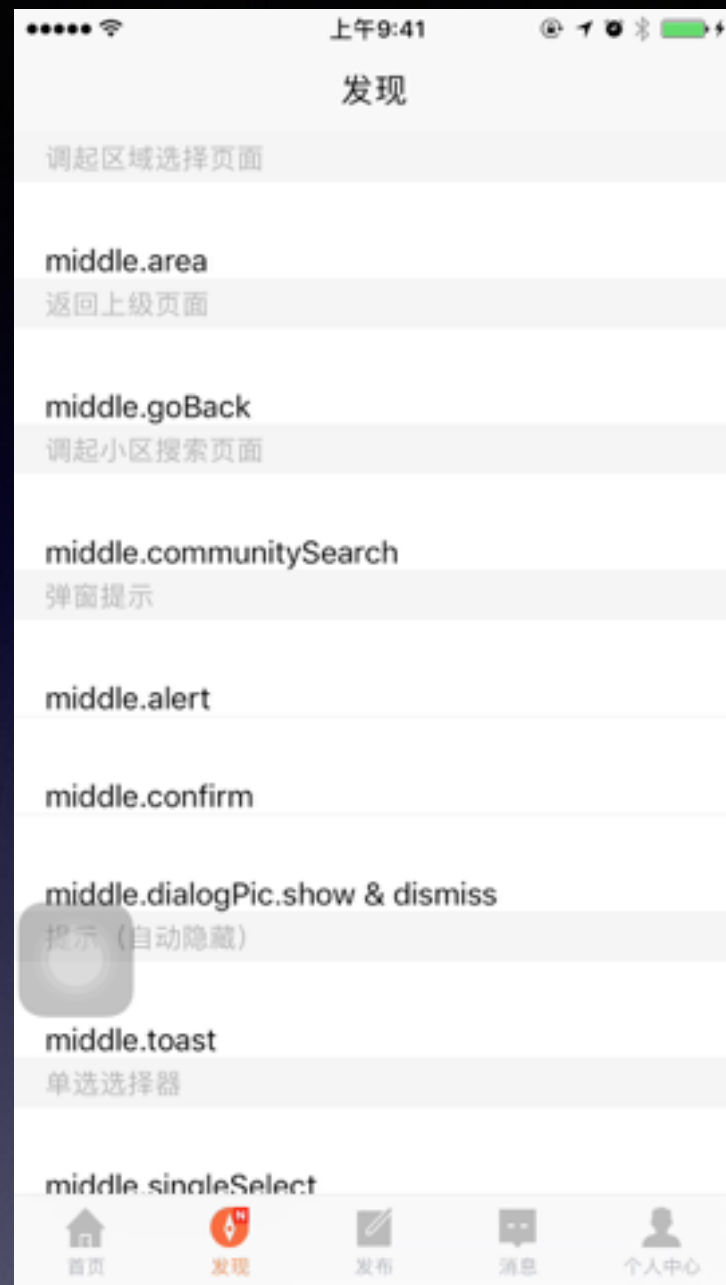
```
/**
 * View路由控制
 */

'use strict';

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

// class HelloWorldApp extends Component {
//   render() {
//     return (
//       <View>
//         <Text>Hello world!</Text>
//         <Text>This is 58 RN world!</Text>
//       </View>
//     );
//   }
// }
//
// AppRegistry.registerComponent('Wuba', () => HelloWorldApp);

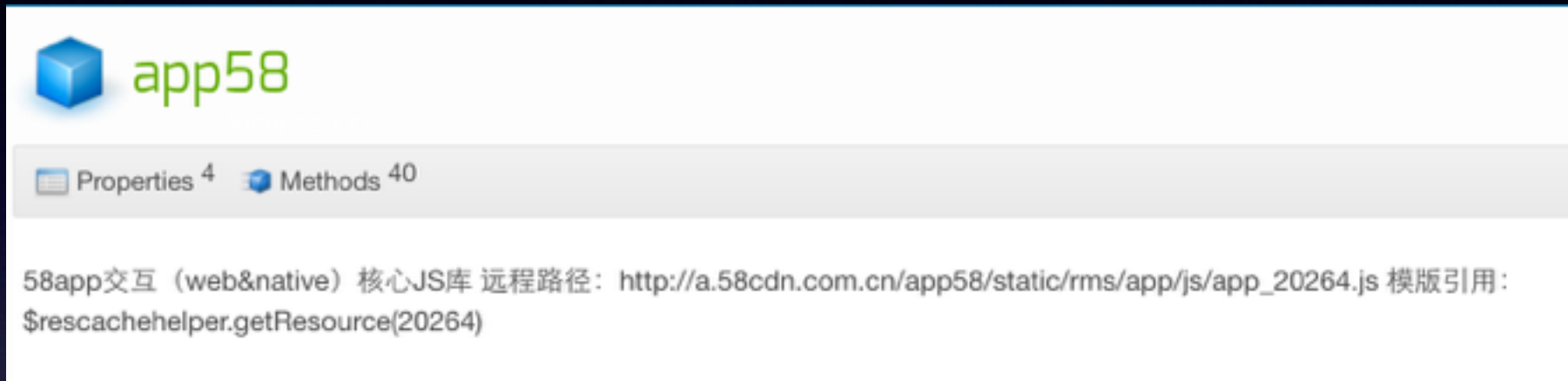
// 下面是中间层demo的例子, 开启需注释15~26行代码
import middle from '../middle/demo';
AppRegistry.registerComponent('Wuba', () => middle);
```



IOS middle demo By 宏伟

3.中间层

web页面－app.js



PS: 顺便推广一下我们的新版app.js

http://apptest.58.com/static/test/app/new_hybrid_frame/docs/index.html

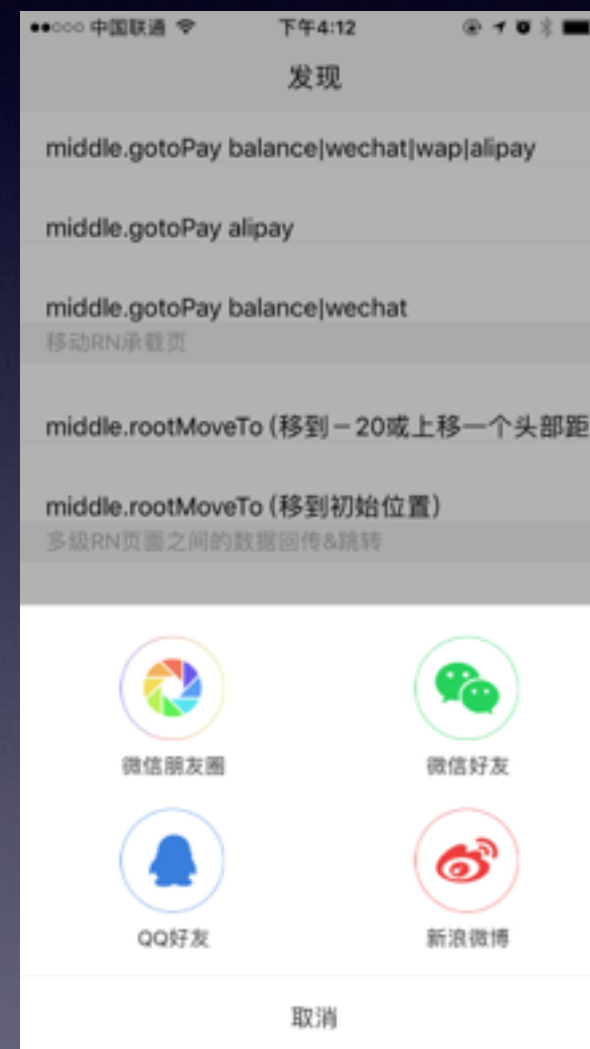
概述

该SDK提供58APP内 js 与 Native 交互的能力，让前端开发者可调用APP的一些原生控件或获取Native存储的数据等。

调用分享控件、页面跳转、发送统计埋点、设置网页标题等，更多功能请[查看Hybrid设计文档](#)

RN页面－中间层

有些native的组件难以实现，需要native来做，我们封装一层，提供给业务线的人使用，这个有点类似于Hybrid里面的app.js的功能



Confluence 空间 人员 创建 ...

▼ [水滴项目]发布React Native化

- 业务架构
- ▶ Native层
- ▼ JS中间层
 - ▼ API路由
 - **area 调起区域选择页面**
 - goBack 返回上级页面
 - communitySearch 调起小区搜索
 - alert 一个按钮弹窗提示
 - getLocation 获取地理位置
 - headInfo 已弃用请使用initialF
 - multiunlinkSelector 多选无关联
 - navigator 页面跳转
 - setWebLog 埋点
 - singleSelect 单选选择器
 - toast 提示
 - voiceRecord 语音输入
 - loading 加载动画
 - photoSelector 图片编辑

页面 /... / API路由

area 调起区域选择页面

创建：雷涛，最新修改：蒋宏伟 于 2016 09 12

API描述

调起小区选择组件

功能截图



09:41 100%

二手车

rootMoveTo 根视图位移到 rootMoveTo 根视图重置

asyncArea 异步区域选择 area 区域选择

on('back') 绑定 off('back') 解绑

中间层的文档

<http://c.58corp.com/pages/viewpage.action?pageId=3997772>

4. 上线

Web

code部署



WinSCP



FileZilla

FCM (RMS)

代码部署到服务器

58APP

RMS平台

通过RMS系统进行上线，进行版本管理
合理利用APP缓存策略

RN

code部署



gitlab















开发代码录入gitlab

58APP

热更新平台

通过热更新平台进行编译过程，并最终生成下发给native可用的patch文件

RN

Name	Last Update	Last Commit > 3f9ed9db – 行高
 App	about 19 hours ago	 jianghongwei 行高
 .gitignore	13 days ago	 leitao #topicgroup
 README.md	2 months ago	 jianghongwei@58ganji.com 话题广场 完成UI
 config.js	2 months ago	 jianghongwei@58ganji.com 计数器 demo
 index.android.js	about a month ago	 leitao diff patch youhua
 index.ios.js	about a month ago	 leitao diff patch youhua
 package.json	about a month ago	 jianghongwei@58ganji.com https 图片

发现项目

Name
App
.gitignore
README.md
config.js
index.android.js
index.ios.js
package.json

发现项目

Name
App
middle
.gitignore
README.md
index.android.js
index.ios.js
package.json

share-rn-demo

真正项目录入git的时候是不需要middle，但本地确实需要

不然，你用不了中间层的方法咯

.gitignore也有写到push的时候忽略middle

```

37 # 本地文件
38 doc/
39 middle/

```

录入热更新平台

http://10.9.192.26:3100/

资源管理

彦祖很忙退出

m资源管理

m资源管理

m资源审核上线

app升级管理

m操作记录

用户管理

rn资源管理>修改资源

返回

静态资源URL(测试环境)

http://gitlab.58corp.com/wuxianfe/gelunbu-rn.git

所属业务线

客户端

用途类型

大类页

描述(资源说明必填):

发现页面哥伦布项目

是否预加载

☐否☒是

是否强制更新

☒否☐是

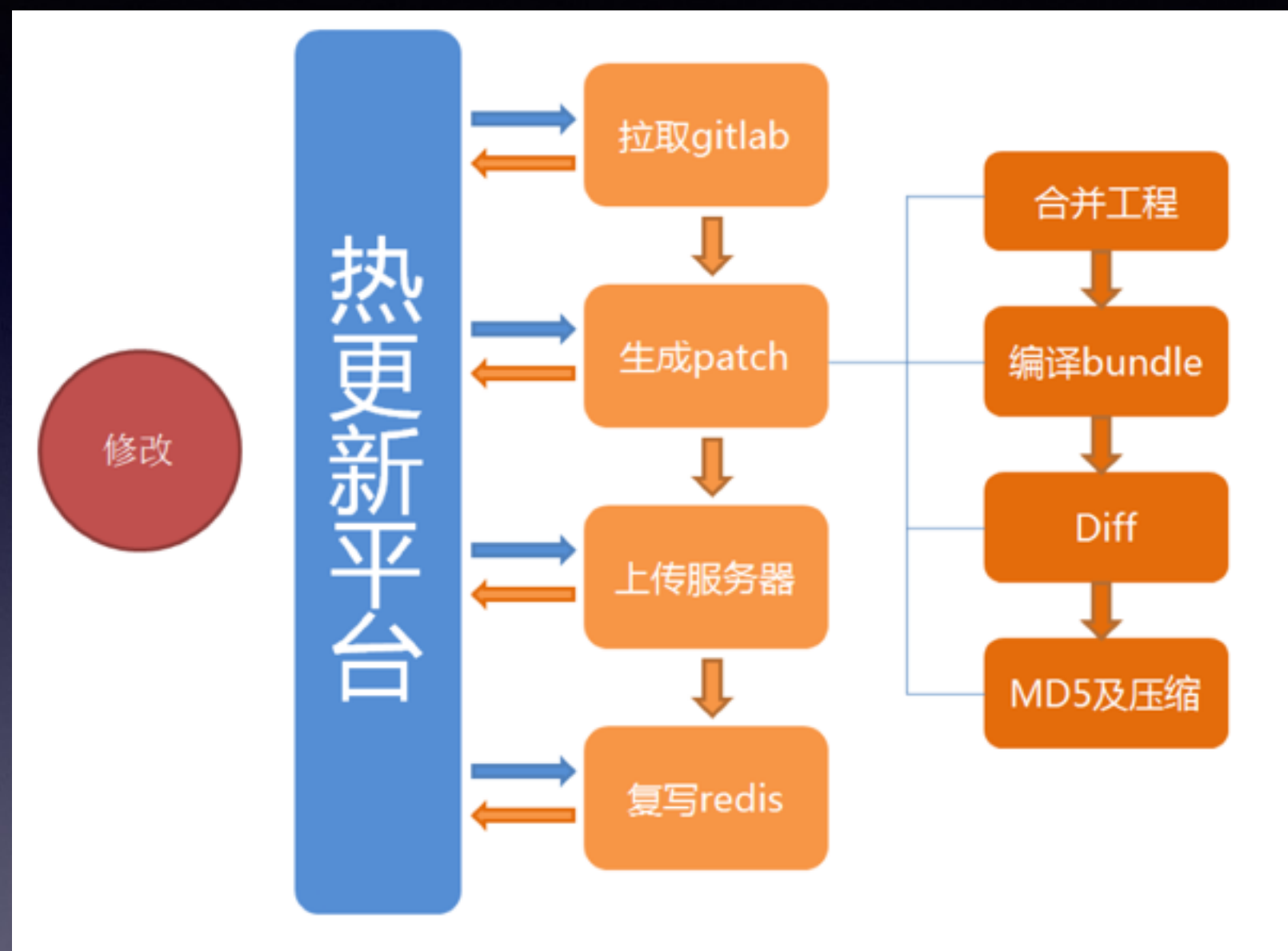
版本号

+1

提交

返回

热更新平台的处理流程



热更新平台返回一个bundleid

RN资源管理							
添加新资源							
<input type="checkbox"/>	操作	资源ID	静态文件URL	版本号	修改日期	创建日期	
<input type="checkbox"/>	修改 上线	5	http://gitlab.58corp.com/jianghongwei/middle2.git	45	2017-02-14 14:38:27	2016-09-09 10:30:53	
<input type="checkbox"/>	修改 上线	11	http://gitlab.58corp.com/wuxianfe/gelunbu-rn.git?ref=	27	2017-02-09 18:39:42	2016-12-14 16:09:53	
<input type="checkbox"/>	修改 上线	12	http://gitlab.58corp.com/wuxianfe/gelunbu-rn.git?ref=	1	2017-02-09 15:50:50	2017-02-09 15:50:50	
<input type="checkbox"/>	修改 上线	7	http://gitlab.58corp.com/wuxianfe/gelunbu-rn.git	102	2017-01-17 14:25:35	2016-11-07 18:49:18	
<input type="checkbox"/>	修改 上线	2	http://gitlab.58corp.com/post-fe/app-rn-refactor.git	61	2016-12-20 14:50:31	2016-06-28 20:26:13	
<input type="checkbox"/>	修改 上线	3	http://gitlab.58corp.com/post-fe/app-rn-refactor.git?	22	2016-12-13 15:02:48	2016-08-16 11:04:12	
<input type="checkbox"/>	修改 上线	10	http://gitlab.58corp.com/wujianyu/app-rn-refactor.git	5	2016-12-08 11:09:28	2016-12-02 17:51:52	
<input type="checkbox"/>	修改 上线	6	http://gitlab.58corp.com/jianghongwei/middle3.git	8	2016-11-24 16:03:44	2016-09-09 10:30:53	

5.跳转协议

web页面

loadPage(pagetype, url, title, showarea, showpub, isfinish, top_right, is_recent, showsift, backtoroot)

调用native页面跳转功能

Parameters

- pagetype : String
必填项，页面类型[childcate|list|detail|link|publish|backreload|mypublish|usercenterbusiness]
- url : String
必填项，页面url
- title : String
必填项，页面标题
- showarea : Boolean
选填项，是否在title上显示区域筛选
- showpub : Boolean
选填项，是否在title上显示发布按钮
- isfinish : Boolean
选填项，跳转到目标页面，源页面是否需要关闭

RN页面

loadReactNative(bundleid, title, url, param)

调用native设置埋点功能，埋点参数需要结合埋点系统预先设置才会生效

Parameters

- **bundleid**: Number
必填项，页面唯一标识，可在热更新平台中查询
- **title**: String
必填项，页面导航显示内容
- **url**: String
选填项，andr平台用来兼容4.0系统（andr4.0系统不支持react native）
- **param**: Object
选填项，透传参数

```
WBAPP.loadReactNative(1, "RN测试页面", "http://p.webapp.58.com/...", {infoid: "1234", cateid: "5"});
```

发现配置的跳转协议

```
{
  "action": "pagetrans",
  "tradeline": "RN",
  "content": {
    "pagetype": "RN",
    "bundleid": "7", // 页面的标示, 来自于热更新平台
    "title": "发现",
    "url": "https://distest.58.com/discovery/home",
    "backtoroot": false,
    "isfinish": false,
    "params": {
      "pagestate": "",
      "infoid": "",
      "cateid": "",
      "catename": "",
      "needlogin": false,
      "type": "",
      "tradeid": ""
    }
  }
}
```

6.总结

开发RN页面的一般流程

1. 独立开发(需要特殊包)
2. 项目上传git
3. 使用热更新平台更新(公司项目才用!!!)
4. 配置入口(跳转协议)

Thanks