

ReactNative 客户端说明：

本文档最后编辑于：2018 - 6 / 20 号

环境说明：我们接入ReactNative 版本为0.47。

1 Rn基础说明

对Rn不了解的可以参考该网站：[ReactNative中文网](#)。

客户端主要给Rn端提供一个可更新和展示的容器，供Rn可以在客户端进行使用。
[ReactNative更新Sdk说明](#)

2 Rn更新名词解释

- 增量更新：

增量更新可以让差分包变得更小，reactNative 框架部分不用再去下载，减少下载的速度。差分包，是根据后台在上传业务包的时候生成的只生成最近五个版本的包，多了以后就生成全量包。差分包的生成和合并是通过bsdiff实现。

- 校验

下载完以后我们会计算md5比对是否正确，如果不正确则会回滚。

回滚就是将在下载的时候备份的bundle文件进行重命名，然后通知界面出错了，用户可以点击重试，然后重新展示界面。

- 强制更新

强制更新指的是这个包，在下载完以后才进行展示。

- 非强制更新

非强制更新指的是这个包下载完以后，下次打开这个界面才会展示。

特殊提示，如果现在客户端已经有包了，则首先进入界面，然后进行更新操作，如果是强制则下载完成以后会进行刷新界面，如果是非强制则下次打开展示新的界面。这样做是为了避免请求更新的时间。缩短用户看到界面的时间。

- ReactNative 渲染白屏问题

因为Rn的渲染时间会比较长在开始渲染的时候会有一个白屏，所以我们为了让体验好点是加了一个小车跑的加载动画。当ReactNative 加载成功以后会通知客户端将小车进行隐藏。

- 业务分包

我们对于每个包都是业务独立，粒度比价小，这样让业务包在每次更新的增量包尽量小。每个业务都有自己的一个package name每次去请求也都是根据这个区请求的，请求的时候会带上自己的本地的版本，这样服务器下发更新数据让客户端可以下载更新。

包的版本信息是存在sp的根据依据package Name进行区分，业务包也是存在data目录。

- 包合并

客户端通过JNI的方式实现了BsPatch 客户端在第一次下载完包以后，会和客户端本地base.android.bundle 进行合并。客户端在更新的时候会和本地存在的业务包进行合并。bsdiff最后生成的差量包是压缩过的，所以无需再压缩。

- Base包版本，业务包版本

Base包版本指的是ReactNative版本对应的版本，我们ReactNative版本不支持热更新，因为这个是跟着客户端走的，如果要更新只能在新包才能正常使用，旧版本的包就没有办法进行上传了。记录在RnConfig类中。

业务包版本指的是某个业务包每次升级以后的版本。记录在sp中，key是包名。

- 增量包

两个业务包之间通过bsdiff生成的差分包。

- 全量包

业务包和基础包（base.android.bundle ）生成的差分包。

3 服务器请求接口说明

[http://xmiles.cn/utis_service/common?
funid=26&shandle=0&handle=0&rd=1522307452632](http://xmiles.cn/utis_service/common?funid=26&shandle=0&handle=0&rd=1522307452632)

请求服务器utils_servce 26 接口。

请求体

```
{
  "handle": 0,
  "shandle": 0,
  "data": {
    "phead": {
      "pversion": "35",
      "phoneid": "251e4c397afde6e",
      "aid": "251e4c397afde6e",
      "imei": "863696037043198",
      "cversionname": "3.36",
      "channel": "1000",
      "activity_channel": "1000",
      "original_channel": 1146,
      "lang": "zh_cn",
      "imsi": "46007",
      "dpi": "1080*1794",
      "sys": "7.0-24",
      "net": "WIFI",
      "mac": "18:D2:76:AF:A0:05",
      "cversion": 336,
      "phone": "FRD-AL00",
      "platform": "android",
      "access_token": "",
      "cityid": "440100",
      "gcityid": "",
      "lng": 113.33371,
      "lat": 23.138901,
      "prdid": ""
    },
    "pkg_name": "violation", // 请求的包模块
    "pkg_version": 0, // 本地业务包版本
    "pkg_base": 2 // 本地base包版本, 对应ReactNative 版本。
  }
}
```

服务器返回结果

```
{
  "update_type": 2, //更新类型分为三种, 0 代表无更新可用, 1 代表是增量更新, 2 代表是全量更新
}
```

```

    "force": true,          //是否是强制更新，如果为true则代表是强制更新。
    "max_version": 62,     //返回最新的业务包版本。
    "download_url":
    "http://rn.xmiles.cn/reactNative/violation_0_android_62_base_2_1529
    373225792.patch", //可供下载的增量包地址
    "md5": "0cb7543b14b91bd7095bdf1de20be019", //下载的包的MD5值，已经
    不使用了。
    "buz_md5": "6f8a3a646f71e0b6f2af2332d586e203", //最新包的MD5值，最
    后本地合成好以后的包会和这个值比对，看是否合成了正确的包。
    "result": {
        "status": 1
    },
    "costTime": 7
}

```

4 ReactNative 版本升级说明

- 为什么要进行ReactNative 版本升级？ 一般情况我们不会对Reactnative 进行版本升级，除非最新版的Reactnative 解决了什么重大bug，或者是性能有很大提升。不升级原因是如果升级，市场上旧版本的安装包无法再更新最新的Rn包，并且会提示用户安装最新的安装包，并且我们的后台在多Base包更新时存在一些问题，所以升级需谨慎。如需升级ReactNative 版本可参考：[ReactNative更新Sdk说明](#) 最后的步骤。

rnandroidsdk说明：

- 因为客户端本地依赖的是我们自己构建的ReactNative 库，该库是基于ReactNative版本0.47 版本构建，我们之所以自己构建ReactNative库是为了解决ReactNative 框架报错的问题，我们对库进行了修改，对有些框架报错，进行了捕捉，然后通过异常捕捉反馈到ui层。然后允许用户进行重试。我们引入的Rn库版本
compile 'com.xmiles.rnandroid:rnandroidsdk:1.0.2' 该项目路径是：
svn://liuzhenghai@company.xmiles.cn:1212/StarBaba/server/npm/react-native-android-modules

5 ReactNative view说明

客户端封装了一个ReactNative view使用方法和ReactNativeFragment使用方法相同，可使用场景是例如在某个Activity中一部分是使用ReactNative 。我们在首页豆腐块哪里支持ReactNative 版的豆腐块。

6 传给ReactNative 端的参数说明

phead	协议头
fullPhead	讯飞广告phead
mobnum	手机号，如果能拿到
debug	是否是debug模式
host	主机地址
moduleName	模块名
title	标题
showTitle	是否展示标题
extra	附加信息

相关类：所有相关类在reactnative 包下。

ReactNativeFragment 展示ReactNative View 的Fragment。

ReactNativeActivity 对ReactNative Fragment 进行包装的activity。

RnUpdateTask 进行ReactNative 更新的Task任务，包括请求，下载校验等流程。

RnBundleMerge 对Reactnative 包进行合并的类。

RnBundlePorvider Rn包管理类，和Rn包版本管理类。

RnUpdateManager 对RnUpdateTask进行管理的类。

7 主要流程图

