

PaiBandRN 打包及热更新协议

web前端打包流程

项目编译脚本配置在工程 `build_conf` 目录下，在根目录执行 `npm run build` 进行编译打包，生成的文件在 `dist` 目录下。

`dist` 目录结构：

`vx.x.x` 版本号对应的目录，打包后生成的ios及android对应的 `zip` 包在此目录下，如

`ios.20170406v1.zip` `android.20170406v1.zip`

`vinfo.json` 当前版本信息描述文件，详细解释见下文

vinfo.json结构描述

```
{
  "buildDate": "4/6/2017, 3:09:28 PM",
  "buildVersion": "v1.3.11", //默认版本号，未特殊指定则加载本版本下的zip包
  "buildNumber": "20170110v3", //默认编译号，未特殊指定则加载本编译号zip包
  "ios": {
    "buildVersion": "v1.3.10",
    "buildNumber": "20170108v1",
    "md5": "4af21f9f0ae9cd0750e07e222cb9f769"
  },
  "android": {
    "md5": "f96a799378a805d14d9041e0613e4123"
  }
}
```

各字段说明：

`buildDate` 最后打包日期

`buildVersion` 当前版本（项目版本号）

`buildNumber` 打包版本（项目版本号下的子版本号，可用于热修复）

`ios` `android` 客户端对应版本及打包的md5信息

客户端加载规则

1. 进入PaiBand后无缓存地请求 `vinfo.json` 文件，获取版本描述信息，并加载本地的资

源 jsbundle

2. 网络后台下载 `vinfo.json` 里指定版本号 `buildVersion` 且对应客户端类型的的zip包，加载路径如 `web`前端打包流程 中描述的。如果在 `ios` 或者 `android` 等客户端类型中特别指定了版本号`buildVersion` `buildNumber`，则优先加载之
3. 下载包后进行md5较验
4. 解压zip资源到各客户端指定的目录（解压后可得jsbundle及图片等资源）
5. 下次打开程序时加载本地更新后的资源

如上面示例 `vinfo.json` 中所示：

- 1.客户端加载 `vinfo.json` ，请求 `http://xxxx.putao.com/paiband_path/vinfo.json`
- 2.得到 `vinfo.json` ，里面 `ios` 指定了版本`buildVersion` `v1.3.10` `buildNumber` `20170108v1` ，则ios版本需要下载的zip包地址
为 `http://xxxx.putao.com/paiband_path/v1.3.10/ios.20170108v1.zip` ，`android`端未特殊指定要加载的版本，则加载默认版本号编译号对应的zip包，地址
为 `http://xxxx.putao.com/paiband_path/v1.3.11/android.20170110v3.zip`
- 3.zip包下载完成后较验md5
- 4.解压资源放至相应目录
- 5.用户下次打开的时候使用新加载的资源