

React-Native打离线包

RN-App项目中js代码和native代码，其中js代码的打包是动态的，打成jsbundle的过程中，js会将自己所需要的所有的文件（这里指你的js业务代码中的require或者import部分）导入到bundle中。

RN-App项目的结构

一个完整的RN-app程序通常包含以下几个部分：

1. native代码部分-objc部分。引入react-native框架，代码中新增的部分是Library部分。大小在2.2M左右，至于打成ipa包增大估计在6M~8M以内。

```
pod 'React', :path => './node_modules/react-native', :specs => [
  'Core',
  'RCTImage',
  'RCTNetwork',
  'RCTText',
  'RCTWebSocket',
  'ART',
  'RCTActionSheet',
  'RCTGeolocation',
  'RCTPushNotification',
  'RCTSettings',
  'RCTVibration',
  'RCTLinkingIOS',
]
```

1. js代码部分-rn代码、依赖的第三方库、业务代码等。这部分代码是通过线上下载的代码，不会打到ipa包中，可以通过热加载的方式从服务端加载。这部分如果采用压缩方式打包，大小不会超过1M。js打成bundle包（可以查看一下bundle包的代码，发现里面全是压缩了的js），然后放到服务器上，下载到APP本地更新。

- 1.js业务代码：这部分代码很少
- 2.react-native代码：这部分代码就是facebook的开源框架的js代码，在打成bundle包时，不用担心所有资源文件都会打包，js采用的是动态打包方式，只会将自身require或者import的部分打入bundle包。
- 3.依赖的第三方库：这里指的是我们的js业务代码可能需要依赖一些优秀的react开源框架，如react-native-navbar这种导航栏第三方库，通过require加载，也会加载到我们的bundle包。有一个要注意的是，如果依赖的第三方库不是纯js代码的话，就不能实现热更新。

1. 图片资源部分。这部分通过打包命令打包时生成asset文件夹。

打包命令说明

```
react-native bundle
Options:
--entry-file Path to the root JS file, either absolute or relative
to JS root [required]
--platform Either "ios" or "android"
--transformer Specify a custom transformer to be used (absolute
path) [default: "/Users/babytree-
mbp13/projects/xcodeProjects/AwesomeProject/node_modules/react-
native/packager/transformer.js"]
--dev If false, warnings are disabled and the bundle is minified
[default: true]
--prepack If true, the output bundle will use the Prepack format.
[default: false]
--bridge-config File name of a a JSON export of
__fbBatchedBridgeConfig. Used by Prepack. Ex. ./bridgeconfig.json
--bundle-output File name where to store the resulting bundle, ex.
/tmp/groups.bundle [required]
--bundle-encoding Encoding the bundle should be written in
(https://nodejs.org/api/buffer.html#buffer_buffer). [default:
"utf8"]
--sourcemap-output File name where to store the sourcemap file for
resulting bundle, ex. /tmp/groups.map
--assets-dest Directory name where to store assets referenced in the
bundle
--verbose Enables logging [default: false]
```

打包步骤

1. cd到工程目录（index.ios.js&package.json存放的目录），执行打包命令。

```
react-native bundle --minify --entry-file ./index.ios.js --platform
ios --bundle-output ./main.ios.jsbundle --assets-dest ./
```

1. 命令执行完生成 `.bundle` 文件
2. 在Xcode中添加 `.bundle` 和 `assets`，必须是 `Create folder references` 的方式。
3. 修改 `jsCodeLocation`。

```
jsCodeLocation = [[NSBundle mainBundle] URLForResource:@"index.ios"
withExtension:@"jsbundle"];
```

1. 另外一种打包命令，程序运行时执行该命令。

```
curl
'http://localhost:8081/Examples/UIExplorer/UIExplorerApp.ios.bundle?
platform=ios' -o main.jsbundle
或者
curl 'http://localhost:8081/index.ios.bundle?
dev=false&minify=true&platform=ios' -o main.jsbundle
```

1. 查看项目中的main.jsbundle或者main.os.jsbundle发现bundle文件里是压缩了的js代码，该部分代码包括 业务js+react-native代码+依赖第三方代码

打包遇到的问题

1. 离线包如果开启了chrome调试，hui访问调试服务器，而且会一直loading不出来。一直提示。。。

```
Loading from pre-bundled file...
```

1. 如果bundle的名字是 `main.jsbundle`，app会一直读取旧的，改名就好了。。。非常奇葩的问题，我重新删了app，clean工程都没用，就是不能用main.jsbundle这个名字。这个问题貌似没遇到过。。。
2. 必须用Create folder references【蓝色文件夹图标】的方式引入图片的assets，否则引用不到图片
3. 执行bundle命令之前，要保证相关的文件夹都存在
4. To disable the developer menu for production builds:

For iOS open your project in Xcode and select Product → Scheme → Edit Scheme... (or press ⌘ + <). Next, select Run from the menu on the left and change the Build Configuration to Release.

1. over

使用 `react-native-webpack-server` 打包

重要，待补充

React-Native发布与热更新

打包完成之后，要实现热更新，还需要在原生代码中做一些逻辑处理，简单流程如下：

1. 首先将上一步输出的main.jsbundle文件放到服务器上，可以达成zip包下载，进一步压缩文件大小。
2. 在App启动的时候，判断RN代码是否有更新，若有更新，则将更新的包下载下来；若没有更新，则不做处理。
3. 有新的更新，在加载RN的入口处，将url指向新的文件地址。
4. 同时，为了保证安全性和下载流量，应将main.jsbundle用密码压缩，在下载下来之后，再进行解压处理。
5. over

React-Native增量更新

待补充