


## 小猴偷米备忘录

### 更新herald-web-kernel-android

1.  命令 **yarn build**，打包好文件在 **./dist** 文件中，需要将这个文件里的内容上传的七牛云。

⚠️ 注意 **index.html** 和 **sw.js** 是否上传成功，原来的 **index.html** 和 **sw.js** 需要更改名称（修改的名字要注意格式）

2. 注意 **版本号** 的修改，需要和 **web-service** 中的 **hybrid-kernel** 的保持一致。

3. 刷新预取

### 更新herald-web-kernel

1.  命令 **yarn package**，打包好的文件在 **./hybrid-dist** 中，需要将这个文件里的内容上传的七牛云。

⚠️ 注意 **index.html** 和 **info.json** 是否上传成功，原来的 **inde.html** 和 **info.json** 需要更改名称（修改的名字要注意格式）

2. 刷新预取

### 打包新的Apk

可以参考 android studio 的官方文档，[🔗 网页链接](#)

### 密钥文件herald-android

⚠️ 注意修改版本号以及名称

Build -> Generate Signed Bundle or APK -> APK -> 选择密钥文件并输入密码 -> 选择 Build Variants 中的 release，并且勾选 Signature Versions 中的所有选项。

### 测试APK是否签发成功

安装旧版的小猴偷米，同时安装新版的小猴偷米，测试是否可以覆盖安装。

### 发布新版的APK

1. 将打包好的 APK 上传的七牛云

2. 修改 herald-web-service 中的 **config.yml** 中的 **version**

3. boss 服务器更新 herald-web-service

### rikumi服务器的mongo启动

```
mongo --port=27881
```

```
show dbs

use admin

db.auth({user:'admin', pwd:'heraldmongo#FKGFW'})
```

## 调试方法简述

### 安卓端

通过andriod studio进行debug，对安卓原生层进行调试。

PC端运行herald-web-kernel-android项目，移动端单击小猴logo十次进入到调试模式。此时必须保证PC端和移动端位于同一个局域网下，监听的端口必须保持一致。此时可以直接输入路由地址切换页面。

通过Chrome浏览器对Andriod app进行调试。将移动端连接到电脑，启动开发者者模式，选择文件传送模式（反正不能选仅充电）（⚠️如果经过多次重试，依旧没有办法识别移动端，可以先打开andriod studio运行一个项目进行配对）。这种调试模式主要是测试各个路由返回的结果的是否正确。

### iOS端

通过xcode进行debug，对iOS原生层进行测试。

打开iOS版小猴偷米的项目文件，修改配置文件，确认网址是否正确，开启debug模式。

同样可以通过Safari对iOS应用进行调试。

## Herald Studio Git&Github使用管理细则

✨现在只有herald-webservice项目使用了这个规则，将逐步扩展到其他项目，⚠️注意规范✨

### 分支

**master**：线上分支，受保护，master分支代码直接部署到服务器上

**dev**：开发分支，受保护

不允许🚫直接commit到这两个分支上面（⚠️**master**只接受来自**dev**分支的**pull-request**）。

### 分支管理策略

当你需要添加新的功能特性（feature）或者进行 bug 修复（bugfix）时，请将 dev 分支的改动同步到本地然后在 dev 分支的 HEAD 处创建新的分支：

```
> git fetch origin
> git checkout dev
> git pull
> git checkout -b <新的分支名称>
```

新的分支名称请按照： <username>/<feature|bugfix>-what-would-you-do 格式命名

之后你可以在该分支上进行代码编写然后正常创建 commit，当你需要将代码推送到 GitHub 时，由于远程目前还没有这个新分支的对应分支，你需要使用不太一样的 push 命令：

```
> git push -u origin <新的分支名称>
```

之后就可以正常使用

```
> git push
```

同时你可以在github上看到自己创建的新分支。

当你完成功能的一部分开发，并且确信该部分可以正常使用的时候，你就可以发起 pull-request 到 dev，然后再由 dev 发起 pull-request 需要首先 merge 到 master 分支。

**pull-request 会被 Traivs CI 进行自动检查**，如果检查不通过则无法被 merge，所以在准备合并之间，请你现在本地执行： yarn lint，**确保无错误输出**。

**Code-Review 策略** 被设置为，每个 pull-request 需要额外 1 人审阅，即发起人需要邀请额外的一个人审阅，审阅无误后才可进行合并。

当你的开发分支被合并到 dev 分支后，请手动删除 GitHub 中的对应分支，你的本地分支可以攒够一定数量之后召唤神龙 批量删除。

**dev 分支到 master 分支的合并同样需要额外一人审阅**，确认无误后才可从服务器拉取，投入正式使用。

（以上部分来源：高老板）

## 讲座数据的导入

✨为了确保导入数据的可靠性，执行数据导入的脚本代码时，**请先注释掉导入代码，控制台打印导入数据，确定数据可靠**

数据导入后需要手动重启服务程序。

⚠️注意及时备份数据库

打卡记录数据

🚫禁止将一份数据重复导入两次。（导入过的数据文件名会自动加后缀.imported）

讲座信息数据

⚠️讲座信息的数据时历年讲座信息的总结，需要手动删除数据库原有的数据后，导入新的数据。

⚠️ excel中的时间格式不是字符串，需要修改格式，**TEXT(<数据源>,"yyyy-mm-dd")**