

## 入职引导：移动开发者专用版

1. [移动组技术栈介绍](#)
  - 1.1 [Cordova](#)
  - 1.2 [原生技术](#)
  - 1.3 [其他](#)
2. [插件](#)
  - 2.1 [插件的创建](#)
  - 2.2 [插件的开发](#)
3. [插件仓库地址](#)
4. [APPCloud平台使用](#)

---

### 移动组技术栈介绍

当前移动组主要围绕名为 `Cordova` 的框架进行日常开发。Cordova框架是hybridAPP的解决方案之一。Cordova产出的APP中，页面主要由webview展示。

针对webview天然存在的功能限制，性能限制等问题，Cordova框架为web页面提供 `插件`，`插件` 具有原生能力,如原生的动画、线程、数据库等。从而对webAPP中体验不好的场景进行支持，使业务团队能将主要精力用在快速推进业务逻辑上。

因此在移动端的日常开发工作中，`插件的开发和维护` 占有较大的比重。

#### Cordova

Cordova框架由Apache提供。在2011年10月，Adobe收购了Nitobi Software和它的PhoneGap产品，Cordova是Apache收购PhoneGap后的一个开源项目，是从PhoneGap中抽出的核心代码，是驱动PhoneGap的核心引擎。可以把它们的关系想象成类似于Webkit和Google Chrome的关系

官网地址：<https://cordova.apache.org> 中文网：<http://cordova.axuer.com>

请按照官网提示，`配置Cordova开发环境`。

#### 原生技术

Android端：旧的版本使用java，新插件使用kotlin

iOS端：主力语言为OC，当前swift语言使用较少

## 其他

1. 除原生Android和iOS语言外，由于Cordova的插件开发工作包含部分 JavaScript 代码的编写，因此会经常使用到 JavaScript 语言。
2. 因插件的开发目的为提供给web端同事使用，并且Cordova项目的运行载体为APP内webview，因此了解一些 web开发 机制， webview 的运行机制等对于日常的开发工作很有帮助。
3. Cordova框架也有其自身的局限，为了持续提升开发效率，提高APP质量，团队在其他语言层面，框架层面均有进行探索和尝试。如React Native(React),Flutter(Dart),原生接入跨端方案等。
4. 维护项目、插件，文档是必不可少的。因此需要文档编写的能力，团队统一使用 markdown 进行文档编写，文档风格可参考已有的插件。

## 插件

插件开发是Cordova开发中必不可少的环节。通过开发插件，可以将大量的原生能力提供给web端使用。

### 插件的创建

1. 搭建Cordova开发环境，详见Cordova官网
2. cd 到 某目标路径
3. 终端执行命令创建插件：plugman create --name [插件名] --plugin\_id [插件id] --plugin\_version [插件版本]

```
// 示例
```

```
plugman create --name testName --plugin_id testID --plugin_version 1.0.0
```

4. cd 到 该插件目录
5. 为插件添加支持的平台，执行完毕后，会在插件目录的src文件夹下出现对应的平台文件夹

```
// 示例
```

```
plugman platform add --platform_name android  
plugman platform add --platform_name ios
```

6. 生成插件的package.json文件：plugman createpackagejson ./  
命令执行后一路回车即可，一般不需要额外配置。

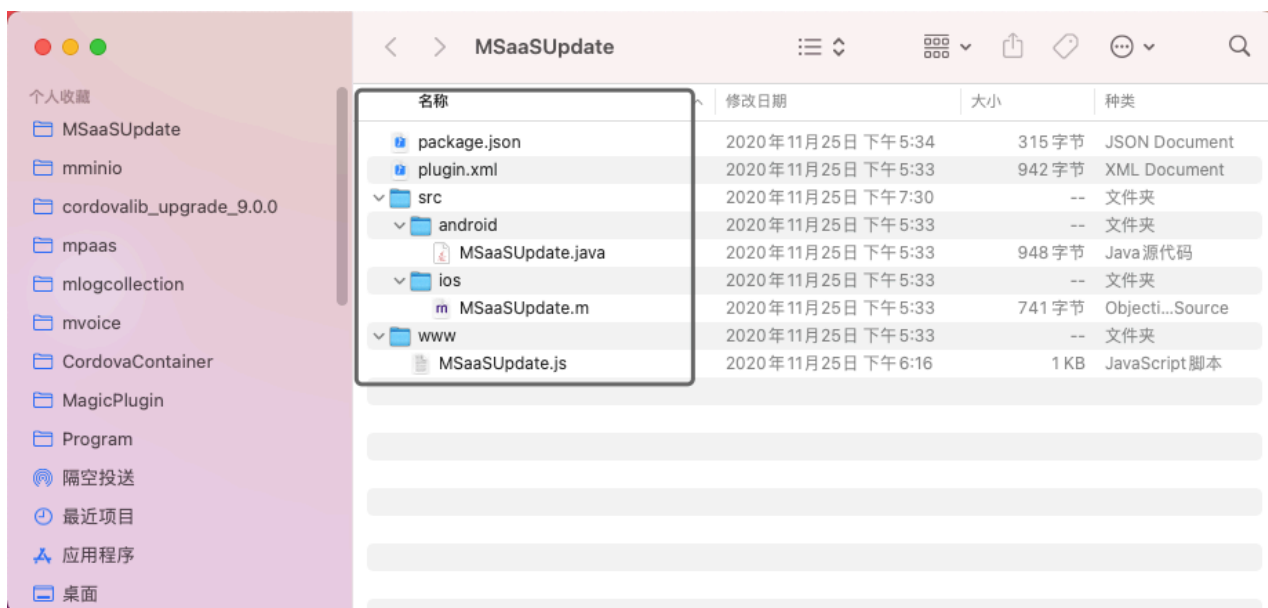
```
// 示例
```

```
plugman createpackagejson ./
```

7. 插件创建完毕。可进入对应平台文件夹中，开始代码开发。

## 插件的开发

新创建的插件，文件路径下包含如下内容：



插件开发过程中，使用较多的文件为 `plugin.xml`，`原生文件` 和 `www/pluginName.js` 文件。

### 开发流程：

1. 与需求方对接需求，将需求转化为插件接口，确定插件的名称，接口数量，命名，字段等内容
2. 创建插件仓库，将1中确定的内容输出为 `README.md` 文档，上传文档到仓库中，后续开发按照文档进行
3. 将插件添加到测试项目中
  - 3.1 cd 测试项目路径
  - 3.2 执行 `cordova plugin add [插件路径]`
4. 插件接口开发：
  - 4.1 在 `pluginName.js` 中，添加新的接口方法、参数、回调函数等。（可仿写插件的示例方法，其他插件的方法，也可根据js语言自行发挥）
  - 4.2 在原生文件中，添加对应的方法，实现该方法，并注意方法回调的使用，避免遗漏（`CDVPluginResult`）。可自行创建需要用到的其他原生文件
  - 4.3 修改 `plugin.xml` 内容，根据实际情况，添加需要引用的原生文件的路径、三方库、项目需要配置参数、钩子等
5. 自测完毕，提交到remote，通知测试同事进行测试。测试通过后由测试同事发版

#### ● tips

##### 1. plugin.xml文件

该文件是插件的配置文件，负责管理插件的引用名，文件资源，hook/钩子，三方包引用，配置各种参数等。内含较多，使用频率较高，因此需要较多的关注。关于 `plugin.xml` 文件的属性含义、使用方法等内容，详情请参考：[https://cordova.apache.org/docs/en/latest/plugin\\_ref/spec.html](https://cordova.apache.org/docs/en/latest/plugin_ref/spec.html)

##### 2. Config.xml文件

使用Cordova框架产出的项目，会包含有 `Config.xml` 文件。该文件内容与 `plugin.xml` 非常相似，它是整个项目的配置文件。部分配置操作需要在这个文件下进行，详情请参考：[https://cordova.apache.org/docs/en/latest/config\\_ref/index.html](https://cordova.apache.org/docs/en/latest/config_ref/index.html)

### 3. hooks/钩子

以上两个文件中，均有hook的概念，我们称之为hook或钩子。指的是自行添加的插件脚本或项目脚本。这些添加的特殊脚本，可以用来自定义cordova命令，执行部分自动化配置或操作等，在开发中也经常用到。详情请参考：<https://cordova.apache.org/docs/en/latest/guide/appdev/hooks/index.html>

## 插件仓库地址

remote地址：<https://gitlab.mypaas.com.cn> 需要权限请联系入职引导人协助开通

该仓库中存放有移动组的全部插件。

## APPCloud平台使用

APPCloud平台地址：<https://appcloud.mypaas.com.cn/#/login> 需要权限请联系入职引导人协助开通

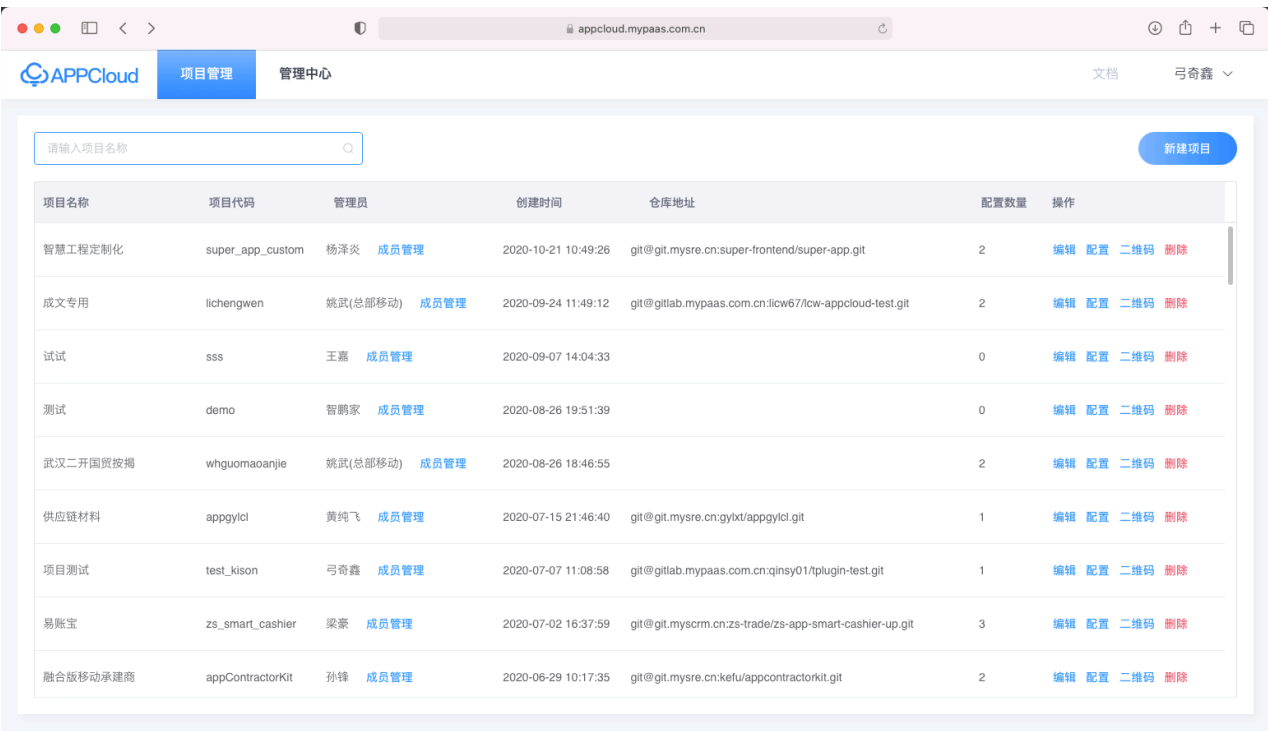
若客户的APP出现bug或其他问题，很多时候我们需要源码来进行调试。通过APPCloud平台可对客户的项目进行分析和构建工程，从而排查问题。

客户项目均可在此平台进行构建。不同项目需要的权限等级不同，因此会有部分项目不可见。

### APPCloud平台使用介绍

介绍内容主要以使用APPCloud平台打包APP，排查插件问题为主

#### 1. 登录后的页面

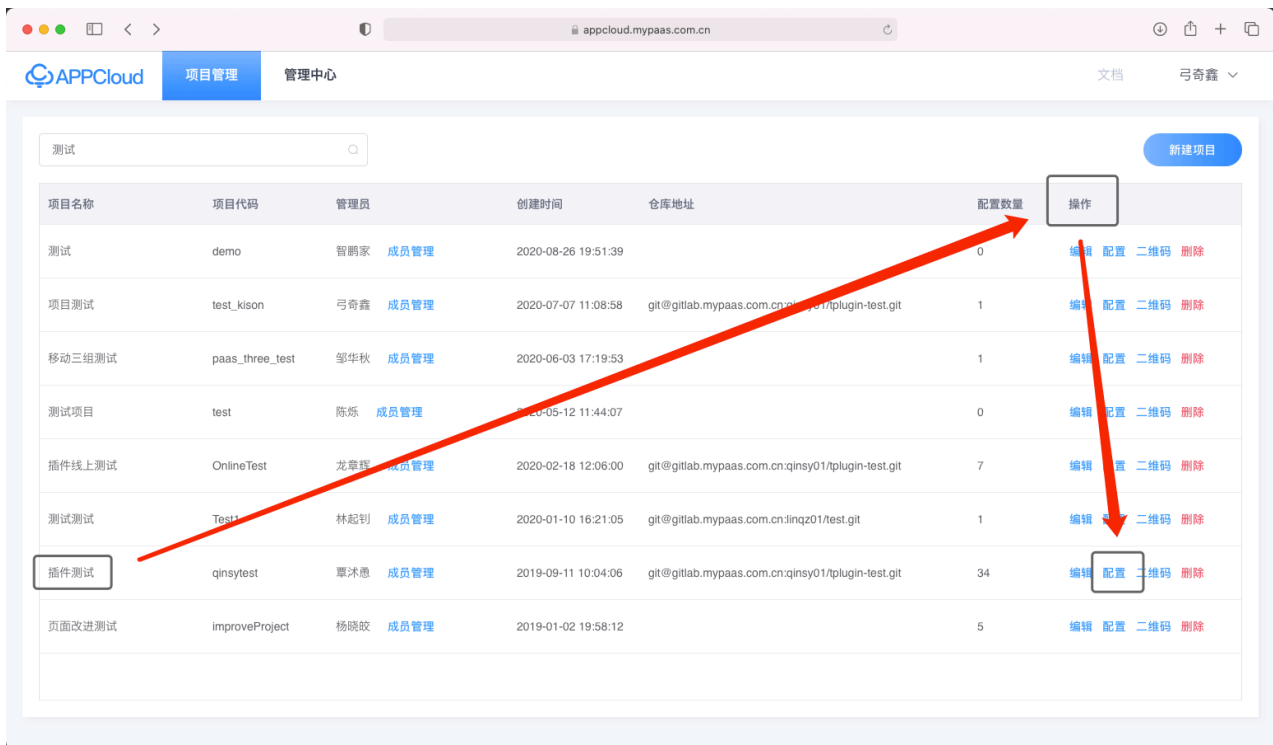


The screenshot shows the APPCloud platform interface. At the top, there is a navigation bar with 'APPCloud' logo, '项目管理' (Project Management), and '管理中心' (Management Center). Below the navigation bar is a search input field labeled '请输入项目名称' (Please enter project name) and a '新建项目' (New Project) button. The main content area displays a table of projects with the following columns: 项目名称 (Project Name), 项目代码 (Project Code), 管理员 (Admin), 创建时间 (Creation Time), 仓库地址 (Repository Address), 配置数量 (Configuration Count), and 操作 (Operations). The table lists several projects, including '智慧工程定制化', '成文专用', '试试', '测试', '武汉二开国贸按揭', '供应链材料', '项目测试', '易赚宝', and '融合版移动承建商'. Each row includes links for '编辑' (Edit), '配置' (Configure), '二维码' (QR Code), and '删除' (Delete).

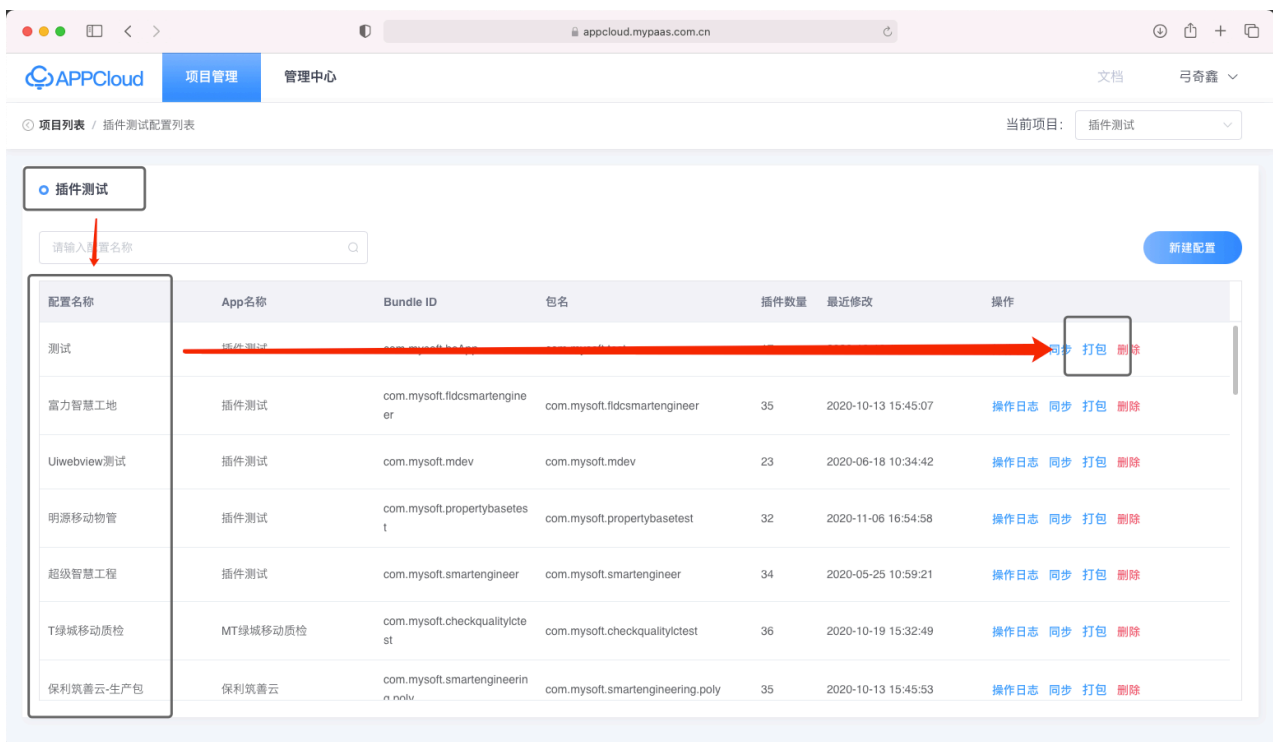
项目名称	项目代码	管理员	创建时间	仓库地址	配置数量	操作
智慧工程定制化	super_app_custom	杨泽炎 成员管理	2020-10-21 10:49:26	git@git.mysre.cn:super-frontend/super-app.git	2	编辑 配置 二维码 删除
成文专用	lichengwen	姚武(总部移动) 成员管理	2020-09-24 11:49:12	git@gitlab.mypaas.com.cn:licw67/lcw-appcloud-test.git	2	编辑 配置 二维码 删除
试试	sss	王嘉 成员管理	2020-09-07 14:04:33		0	编辑 配置 二维码 删除
测试	demo	智鹏家 成员管理	2020-08-26 19:51:39		0	编辑 配置 二维码 删除
武汉二开国贸按揭	whguomaanjie	姚武(总部移动) 成员管理	2020-08-26 18:46:55		2	编辑 配置 二维码 删除
供应链材料	appgyicl	黄纯飞 成员管理	2020-07-15 21:46:40	git@git.mysre.cn:gyxt/appgyicl.git	1	编辑 配置 二维码 删除
项目测试	test_kison	弓奇鑫 成员管理	2020-07-07 11:08:58	git@gitlab.mypaas.com.cn:qinsy01/plugin-test.git	1	编辑 配置 二维码 删除
易赚宝	zs_smart_cashier	梁豪 成员管理	2020-07-02 16:37:59	git@git.mysrcm.cn:zs-trade/zs-app-smart-cashier-up.git	3	编辑 配置 二维码 删除
融合版移动承建商	appContractorKit	孙锋 成员管理	2020-06-29 10:17:35	git@git.mysre.cn:kelfu/appcontractorkit.git	2	编辑 配置 二维码 删除

2. 选择某工程进行打包。这里以名为 插件测试 的工程为例,点击 操作 目录下的 配置 项,进入到该项目下

## 的不同配置项

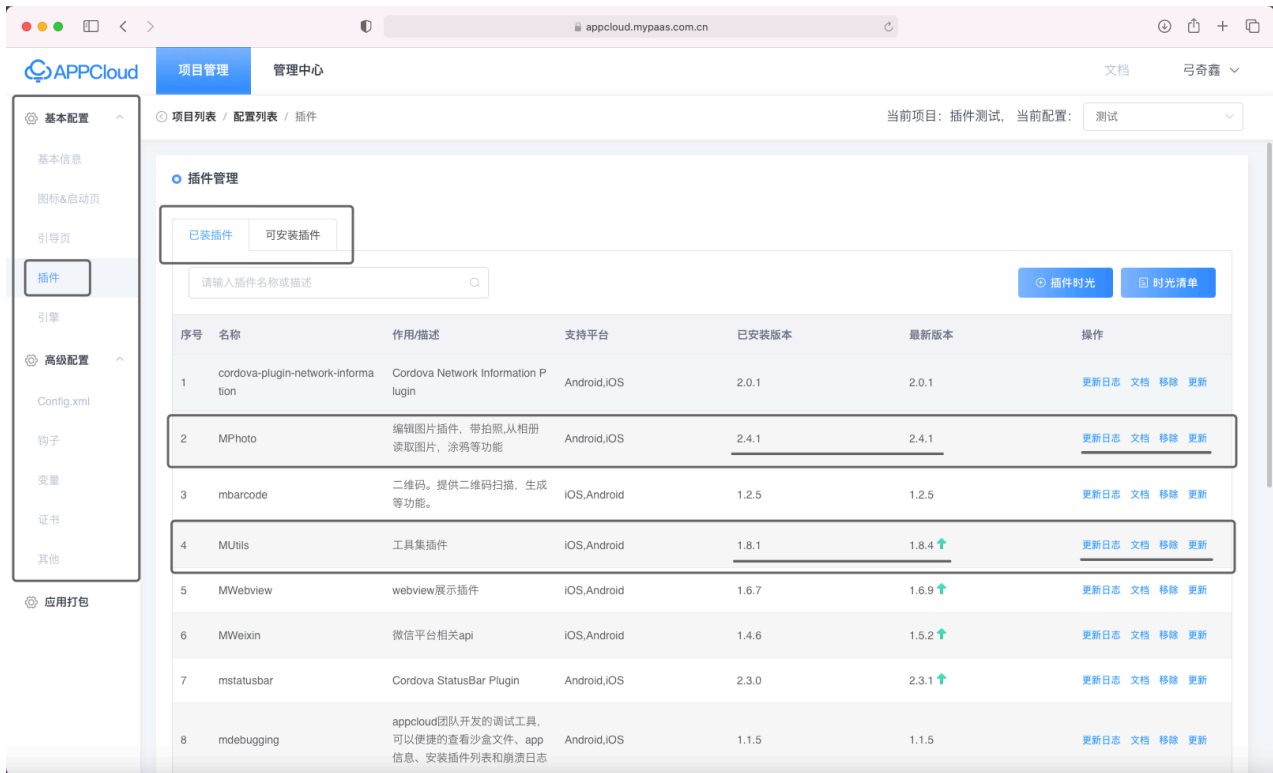


3. 选择需要的配置项。这里以 测试 配置为例，点击 打包，进入到打包页面



4. 打包页面可分为三个板块，分别是包的 配置参数， 打包参数， 和 打包记录

- 配置参数，包含 基本配置 和 高级配置 两个内容。通过查看这些内容项，可以看到该APP的基本信息，部分静态内容，配置参数，证书等。其中 插件 选项，可以查看该APP已经安装的，和可以安装的全部插件，包含插件的版本信息，更新日志等，可以对插件进行升级，新增和移除等操作。插件选项在日常排查工作中使用较多。



- 打包参数，即打包时需要填入的参数信息，根据提示输入即可。其中，MDebugging 开关用于开启 APP本地的调试工具，建议打开。备注内容需要注意：若要获取APP打包后的源码，则必填 appcloud\_debug 字段,不填或填其他内容，打包后源码会自动删除。以上内容配置完毕后，点击开始打包即可。此时可以在打包记录中看到此次打包的记录，会展示为loading状态，并在企业微信群中收到开始打包的提示，该提示会展示打包服务器的IP地址，打包发起人和项目信息等，这些信息可用于后续拉取源码。

#### ● 打包参数

\* 打包版本

资源包地址  该字段填写后，git tag字段将会被忽略。

git tag  [刷新tag](#) last commit: [7da94a3](#) 2020-12-07 18:04:23 覃沐惠 fix

打包平台  Android  iOS

MDebugging

环境变量  dist

备注

- 自己构建APP时，在构建开始时会有如下的开始提示，告知构建使用的服务器地址。
- 其他同事和客户构建失败时，均会在群内通知构建失败消息，方便排查失败原因。

[10.8.1.21] 老爷，您的 [ios] 包在这台服务器上构建

===== 应用信息 =====

A明源移动质检 [com.mysoft.checkqualityalpha]

===== 项目配置 =====

明源移动质检 - Alpha包

===== 我是友好的结束线 =====

@龙章辉

星期三 下午 1:03

别怕，有我 BOT

iOS 构建失败了哟

===== 应用信息 =====

A明源移动质检 [com.mysoft.checkqualityalpha]

===== 项目配置 =====

明源移动质检 - Alpha包

===== TA在构建 =====

龙章辉 18620690634 移动一组 (天际开放平台)

===== 查看日志 =====

[http://appcloud.mypaas.com.cn/api/project-build/build-failed-log?build\\_id=39f933f7-d57f9972-87dd-1788257b245a](http://appcloud.mypaas.com.cn/api/project-build/build-failed-log?build_id=39f933f7-d57f9972-87dd-1788257b245a)

===== 我是友好的分界线 =====

ProjectId: 39eae6f5-e88ff479-4fe3-b75cc5bfeb77

ConfId: 39eae8d9-3e16c4ca-02da-baed5995202



- 打包记录.该内容展示了历史打包的信息，最新的打包信息会被展示在顶部。可以根据平台，打包状态等查找想找的包。在操作选项下，可以查看对应包的插件信息，使用同样的配置再次打包，打开已打完的包的安装二维码等。使用时需要留意状态和操作人选项，因可能存在同一个项目，多个同事同时在打包的情况，若配置相同则尽量使用同一份代码，减小打包服务器压力。

#### ○ 打包记录

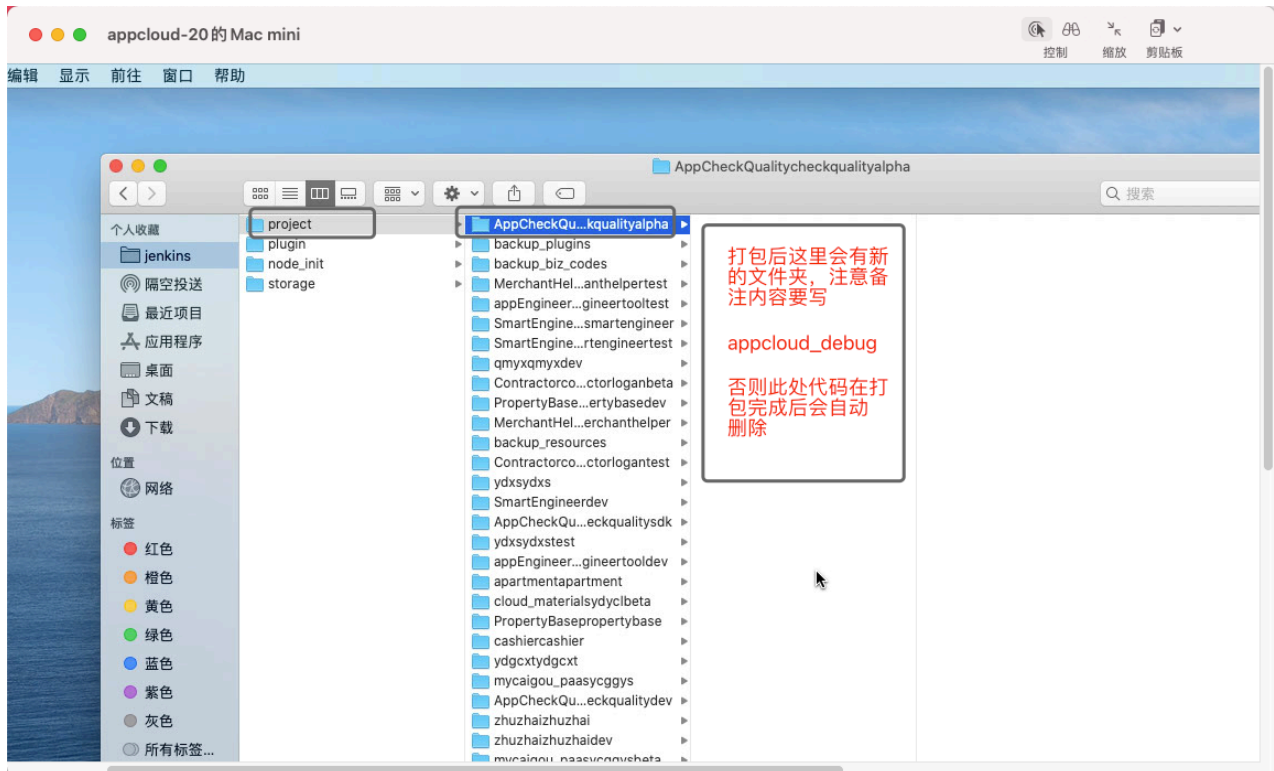
App名称	平台	类型	版本	配置名称	环境变量	git tag	状态	操作人	备注信息	开始时间	结束时间	操作
2c测试包	Android	debug	1.0.3	b2c测试包		not git	● 单沐逸			2020-11-02 19:39:28	2020-11-02 19:40:49	<a href="#">使用该配置构建</a> <a href="#">下载安装二维码</a> <a href="#">插件信息</a>
2c测试包	Android	debug	1.0.3	b2c测试包		not git	● 单沐逸			2020-11-02 19:25:15	2020-11-02 19:26:53	<a href="#">使用该配置构建</a> <a href="#">下载安装二维码</a> <a href="#">插件信息</a>
2c测试包	Android	debug	1.0.3	b2c测试包		not git	● 单沐逸			2020-11-02 19:23:48	2020-11-02 19:24:20	<a href="#">使用该配置构建</a> <a href="#">插件信息</a> <a href="#">查看错误日志</a>
测试	Android	debug	0.0.6	测试包		not git	● 单沐逸			2020-10-30 15:32:24	2020-10-30 15:34:01	<a href="#">使用该配置构建</a> <a href="#">下载安装二维码</a> <a href="#">插件信息</a>
测试	Android	debug	0.0.5	测试包		not git	● 单沐逸			2020-10-30 15:28:24	2020-10-30 15:29:46	<a href="#">使用该配置构建</a> <a href="#">下载安装二维码</a> <a href="#">插件信息</a>

- 拉取源码。打包结束后，在Safari中输入 `vnc://{打包服务器IP}`，回车，即可通过屏幕共享功能，链接到打包服务器。登录服务器时需要输入密码。

```
// 示例
```

```
vnc://10.8.1.20
```

项目代码会存储在jenkins目录下，进入该目录下寻找即可。找到后可以直接拖拽到自己电脑上进行调试。注意：打包成功后没有通知，因此自行打包时需要关注打包的进度和状态。打包记录中 **状态** 选项下，非loading状态即为打包结束，可以拉取源码。(打包失败可能由于服务器上的证书/描述文件过期等原因引起，不影响源码的情况下，打包失败源码也可以使用)。



拉取源码后可进行本地调试，排查问题。如非必要，避免在打包服务器上调试问题。