# 文件说明



游戏启动的时候，通过http拉取远端配置信息，拉取的内容就是apps.json文件的内容。具体过程可看下ResUpgrade.cs文件的OnLoadRemoteConfigEx函数。

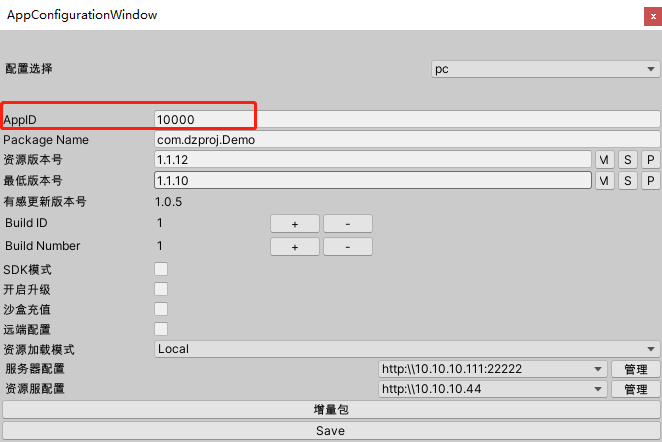
该文件的内容由客户端开发人员写好后，交给后台同学刷新到外网服务器上。

# 文件内容说明

目前文件有四个字段的内容，下面分别说明。

## “apps”

apps字段下包含各个AppID对应的配置。AppID在Unity编辑器中设置。具体如下：



每个appid的内容说明如下：

“store\_url”: 整包更新所跳转的商店地址

“Build ID”：对应上面截图的Build ID，用于区分不同的打包版本。每次更新整包需要网上累加。

“status”: 包状态（auditing, production, development），不同状态对应不同的服务器配置环境；

“is\_open\_upgrade”: 是否启动热更新功能；

“is\_open\_sdk”: 是否开启第三方登录，false就是手机号登录；

“is\_open\_sanddbox”：是否开启沙箱测试账号，用户测试支付功能；

## “servers”

设置不同状态下对应的服务器地址。

“res\_url”: 资源服地址，热更新等资源下载需要用到。

“res\_url\_backup”: 资源服备用地址。

“login\_hosts”: 登录路由服务器。

## “langs”

多语言配置，包含“ios”，“android”和“windows”字段，理论上每个字段下的内容都是一样的，每个平台字段对应的内容来自各平台生成的多语言文件对应的lang\_list.json文件。即将lang\_list.json文件的“langs”字段内容，复制到“ios”、“android”、“windows”字段下。

## “version”

版本号字段，同样区分不同平台。字段含义如下：

“appVer”：程序版本号，目前没用上。

“resVer”：资源版本号，当前使用这个版本号区分不同版本。

“lowestResVer”：最低版本号，本地版本小于最低版本时，需要强制更新整包。

“sentientResVer”：有感更新，目前没用上，如果设置的话，热更前会提示是否更新。

“assets”：包含最低版本到最新版本之间，所有版本所需要下载的对应的zip包。

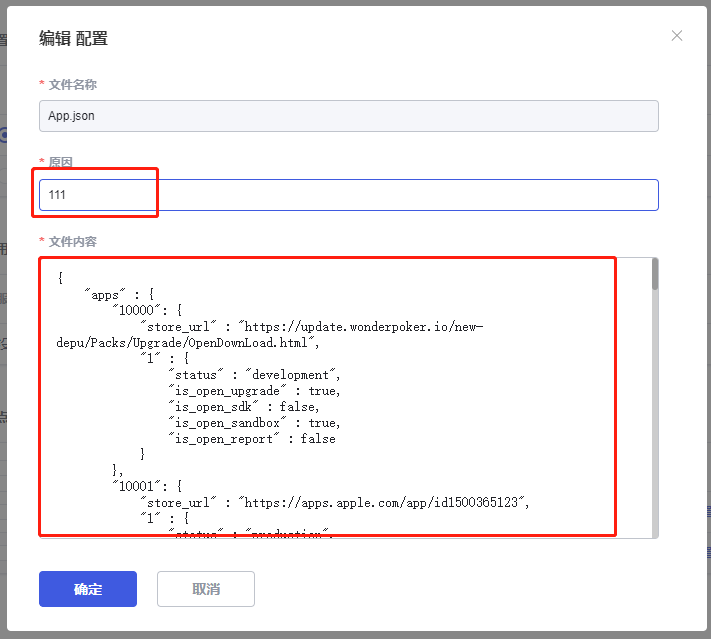
# 常用操作

## 刷新内网配置

开发中比较经常更改多语言翻译，因此会经常修改“langs”字段下的内容，修改完后，还需要将内容更新到内网服务器上。具体操作如下，拷贝整个文本内容，进入以下地址（对应111服务器）：<http://10.10.10.11:3000/index.html#/server/1XPassport.5HttpServer/config>



点击编辑，打开编辑窗口：



将拷贝的最新内容替换原本的文件内容，写上原因，点击确定，就修改好了配置。最后需要重启下服务。

如下图所示，依次点击服务管理，勾选所有服务，点击批量重启，最后会提示重启成功，就表示配置刷新上去了。此时可以到浏览器打开：<http://10.10.10.111:22222/version> 验证下配置是否刷新成功。



82服务器的配置过程一样。地址：

<http://10.10.10.82:3000/index.html#/server/1XPassport.5HttpServer/manage>

## 更新版本

每次更新版本，需要修改version下的版本号和资源文件信息。