

## 1.files 文件生成

### 1.files.txt 文件作用

files.txt 是一个索引文件，里面存放了 lua 文件的路径以及 MD5 校验值。

客户端进行热更新时，连接服务器，首先下载的就是服务器端的 files 文件，然后和本地的 files 文件进行数据对比，检查每一个 lua 文件的 MD5 校验值，如果发现有不同的，就下载相应的 lua 文件，覆盖本地对应的 lua 文件。

files.txt 文件内容：

```
luaHotfix.lua.txt|fcfe491d9aea719aac6c6361b73a7495
```

### 2.编辑器扩展生成 files 文件

- ① 创建一个编辑器扩展脚本，定义一个方法，用于生成 files 文件；
- ② 通过 IO 技术往 StreamingAssets 文件夹下创建一个 txt 文本文件，并写入测试数据；
- ③ 每次生成文件都检查原文件是否存在，如果存在先删除，再创建。

## 2.MD5 算法介绍

### 1.md5 简单介绍

md5 是一种“不可逆”的加密算法：原始数据-->MD5-->加密数据。

md5 加密过的数据会生成 16 位或者 32 位的“数字字母混合排列的字符串”。

123456 经过 MD5 加密后的效果：

16 位：49ba59abbe56e057

32 位：e10adc3949ba59abbe56e057f20f883e

MD5 在线加密解密网站：<http://www.cmd5.com/>

## 2.md5 常见案例

### ①网站用户密码-->加密用户密码。

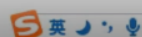
网站注册的时候，需要输入用户名和密码，用户名是公开的，密码才是该用户的保障。

用户名和密码都是存储到数据库中的，比如 MySQL 数据库。然后用户可以使用用户名和密码进行登录。

为了更好的保护用户的密码信息，用户注册的时候，写入到数据库中的密码是需要经过加密的，是绝对不可以存储明码的。比如：

123456-->MD5-->49ba59abbe56e057

或者还可以更强化，比如：用户密码+特定字符串 -> MD5 -> MD5 值。



或者还可以更强化，比如：用户密码+特定字符串 -> MD5 -> MD5 值。

然后用户登录的时候，用户会输入用户名和密码，这个密码还是会走固定的逻辑，计算出相应的 MD5 值，然后以用户名和密码组拼出 SQL 语句，去数据库中查询，有数据，则登录成功，否则登录失败。

### ②文件校验码-->校验文件的真实性，完整性。

常用于下载网站，尤其是软件的官方下载网站。

官网软件下载页面，会有下载地址，以及这个软件安装包的 MD5 值，当用户通过其他途径，下载了相应的软件安装包后，可以用 MD5 软件来计算这个安装包的 MD5，然后和官方公布的 MD5 值进行比对，如果完全一样，表示该安装包是真正的官方安装包，如果发现不一样，这个安装包就是一个危险安装包，可能已经被第三方人员处理过了。