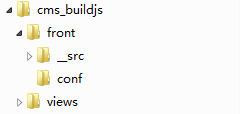
# Buildjs API Documents v1.2

By pakinguo

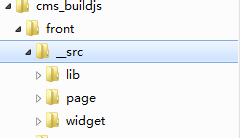
1.2版本主要简化了初始化项目的脚本，把繁琐的脚本执行路径以及命令缩略为更简洁明了的操作。脚本执行方式类似node或者fis，命名为buildjs，如：buildjs -init。下文详细介绍项目目录结构、默认配置文件值和主要相关命令。

## 严格默认的目录结构规范

由于项目设置了默认配置文件，如果没有对配置文件进行修改，则按照默认的配置进行文件监听同步和国际化，此时需要约定默认的文件目录结构，如下图：



以cms为例，front下的\_\_src（两个“\_”）和conf为开发编码时的开发环境目录，其中\_\_src目录结构如下：



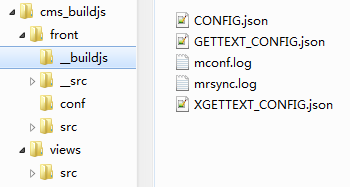
其子文件夹结构与之前所制定的目录结构规范一致。（目前cms中没有page文件夹，取而代之的是business，建议改为无语义歧义的“page”，其代表意思是所有模块都是“页面”下的模块）。

而conf下的文件默认如下：



|  |  |
| --- | --- |
| \_\_conf.js | 生成的seajs配置文件conf.js的框架模板 |
| \_\_jsalias.json | Seajs模块别名alias配置 |
| \_\_cssalias.json | Seacss模块别名alias配置 |
| \_\_ignore.json | 生成发布版本时，进行合并压缩需要忽略的别名alias配置 |
| conf.js | 由\_\_conf.js、\_\_jsalias.json和\_\_cssalias.json一起生成的seajs配置文件 |

初始化项目后，会在front下生成一个目录\_\_buildjs，用来放置相关的编译配置文件，以及初始化front - > src和views -> src（即是之前版本同步脚本所执行的动作）：



\_\_buildjs生成的文件对应功能如下：

|  |  |
| --- | --- |
| CONFIG.json | 生成seajs配置文件、同步文件、将htpl模板转换为seajs模块这三个功能的配置文件，具体配置可以直接到173的测试文件下查看：  173：/data/ivy/ipick/cms\_buildjs/front/\_\_buildjs |
| GETTEXT\_CONFIG.json | 国际化翻译配置文件，同前版本配置，字段有修改 |
| XGETTEXT\_CONFIG.json | 国际化提取待翻译字段配置文件，同前版本配置，字段有修改 |
| mconf.log | inotifywait监听的生成seajs配置文件的文件夹路径，即是front/conf的路径，可通过停止脚本来查找对应的inotifywait的pid并kill进程 |
| mrsync.log | 同mconf.log，用于同步文件和转换seajs模块的功能 |

**备注：由于inotifywait会监听到所有文件夹的变动情况，故需要将生成seajs配置文件的监听与同步文件和转换seajs模块的文件夹监听分离开来，用两个inotifywait进程来完成独立监听，由于停止监听需要查找对应进程pid，故需要先将已监听好的文件夹做记录，方便停止进程。**

## 默认配置文件值&设定

**默认配置文件请参考测试项目：**

173：/data/ivy/ipick/cms\_buildjs/front/\_\_buildjs，配置项与前面的版本相差不多。

**如何更改？**

由于并没有提供任何脚本用于修改配置文件，请对生成的几个配置文件直接vi修改或上传文件覆盖。

## 核心命令&用法

命令源文件位于buildjs文件夹下的bin/buildjs，在linux下使用前，需要将其添加到全局环境变量中：

**ln -s /home/pakinguo/buildjs/bin/buildjs /usr/local/bin/buildjs**

添加到全局环境变量后，就可以切换到需要监听的项目文件夹下，执行初始化脚本。例如，需要对项目路径为/data/ivy/ipick/cms\_buildjs/front进行初始化，则切换到该路径下，再执行命令：

**buildjs -init**

该命令执行后，会在当前文件夹下生成\_\_buildjs文件夹，放置初始化的配置文件，具体相关文件可参考上文。同时，它会执行初始化事件：1、同步\_\_src的静态资源文件到当前路径下的src文件夹（新建）；2、同步\_\_src的.tpl等H5页面模板文件到其上一级的views文件夹下的src文件夹（新建）；3、根据当前文件夹下的conf文件夹里文件生成conf.js的seajs前端配置文件；4、对\_\_src和conf文件夹分别进行inotifywait实时监听。只要\_\_src或conf的相关监听文件有变动，就会对应执行同步脚本和配置文件生成脚本，同时默认会从front的src和views的src对应生成一份国际化英文版本（生成对应的en文件夹爱）。

命令列表及功能如下表：

Usage: buildjs [options] [argv1] [argv2] ...

|  |  |
| --- | --- |
| options | |
| -init [src] | 初始化buildjs功能，需要把当前路径切换到源项目文件夹，如（后续介绍的options使用方式一致）：  ~: cd /data/ivy/ipick/cms\_buildjs/front  /data/ivy/ipick/cms\_buildjs/front: buildjs -init  目录结构建议按照**[严格默认的目录结构规范](#_严格默认的目录结构规范)**，否则需要手动更改生成的配置文件，并手动执行buildjs -wstart重新监听项目文件夹。  参数[src]，可选，若指定src，则以src为准，忽略当前文件夹路径，如：  ~: buildjs -init /data/ivy/ipick/cms\_buildjs/front  等同于上述所执行的。 |
| -wstart [src] [conf] | 启动inotifywait实时监听同步脚本和生成配置文件脚本，该脚本会先kill原有的inotifywait进程，再重新开启新进程。启动后，可通过ps -a | grep inotifywait来查看启用的进程。  参数[src]，可选，同-init的src。  参数[conf]，可选，指定监听时传入的配置文件，即可替代\_\_buildjs/CONFIG.json这个配置文件。 |
| -wstop [src] | 停止inotifywait实时监听功能。  参数[src]，可选，同-init的src。 |
| -xgettext [lang] [conf] | 提取待翻译的词条。  默认是以\_\_buildjs/XGETTEXT\_CONFIG.json配置值为准，从\_\_src里提取所有已标记的待翻译词条，标记格式同之前文档所提到的。  参数[lang]，指定提取出的po文件语言类型，不指定则默认为en，会输出到\_\_buildjs/i18n/i18n.en.po文件，目前仅支持po格式。  参数[conf]，即代替XGETTEXT\_CONFIG.json的配置文件路径。 |
| -gettext [lang] [conf] | 根据已翻译好的po文件翻译已标记的待翻译词条。  会从front的src和views的src同步翻译到指定语言文件夹，如输出到en文件夹，则front和views下都有en文件夹，默认是以\_\_buildjs/GETTEXT\_CONFIG.json配置值为准。  参数[lang]和[conf]同-xgettext。 |
| -release | TODO |
| -v | TODO |

## 总结使用步骤：

**STEP1:**

按照**[严格默认的目录结构规范](#_严格默认的目录结构规范)  
STEP2:**

切换到项目front文件夹再执行初始化：

~: cd /data/ivy/ipick/cms\_buildjs/front

/data/ivy/ipick/cms\_buildjs/front: buildjs -init

就可以尽情开发了。

**STEP3:**

需要生成某个语言的国际化版本：

~: cd /data/ivy/ipick/cms\_buildjs/front

/data/ivy/ipick/cms\_buildjs/front: buildjs -xgettext cn

对i18n.cn.po进行翻译（已翻译过的可以直接保留，提取的会保留翻译的词条，同时去重）

最后执行：

/data/ivy/ipick/cms\_buildjs/front: buildjs -gettext cn

**STEP4:**

发布，1.2.1版本再开发完善。