# Buildjs API Documents v1.1

By pakinguo

## 优化实时同步功能

**主要优化了文件实时同步的机制和效率，对应的配置文件有部分调整。**

新增build-inotify，功能文件如下：

|  |  |
| --- | --- |
| run.sh | 触发inotifywait——实时监听文件变动，并根据变动的目标文件做对应处理，可选参数为run.conf配置文件，./run.sh xxx.conf，默认为build-inotify文件夹下的run.conf文件 |
| run.conf | run.sh的配置文件，只需设定两个值：   |  | | --- | | run.conf:  SRC=/data/ivy/ipick/cms\_4\_inotify/front/\_\_src  BUILDCONF=./CONFIG.json |   SRC为监听的项目源文件  BUILDCONF为exec执行时的配置文件，与未优化版的build-realtime的配置文件一致 |
| init.js | 执行run.sh后一开始触发的时间，这里是做一次全局文件同步 |
| exec.js | 文件变更时执行的相关功能，目前已优化的功能有：  除view层的文件同步，采用rsync来同步已做修改的对应的文件，效率比用nodejs执行的完全同步功能快近10倍；  判断变更文件的类型，执行对应的操作（.htpl的文件类型执行htpl2js的功能，.tpl执行view层同步功能，.json文件更改的执行生成配置文件的功能，其他则为rsync同步）；  未优化功能：针对单一文件的变更做相应的单文件操作（目前都是批量操作） |
| kill.sh | 运行run.sh后，inotifywait会占用一个进程，脚本设计是将进程id存进默认文件（路径为./\_\_inotify\_pid/log.pid，以当前所在目录pwd为相对路径根目录生成的，建议按项目为单位，建立一个配置文件夹并在该文件夹下运行run.sh），通过kill.sh可以直接kill掉该同步脚本进程。 |
| CONFIG.json | 同build-realtime的CONFIG.json |

使用方式：切换到相关项目的配置文件夹下，再执行run.sh，如：

~: cd /home/pakinguo/buildjs/projects/ipickcms\_test

~: /home/pakinguo/buildjs/node\_modules/build-inotify/run.sh ./run.conf

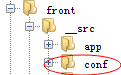
确保在ipickcms\_test下已经有run.conf配置文件，否则会默认加载build-inotify下的run.conf文件。执行之后，会在ipickcms\_test下创建\_\_inotify\_pid文件夹，存放该

## 建议

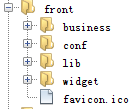
由于inotifywait实时监听的根路径的各类文件变更动作都会监听到，故需要把之前提到的conf目录都整合在同个监听路径下，否则无法实时生成修改后的seajs配置文件，有两种修改方式：

* conf放置到\_\_src

->



* 从front到resource/src，同时front目录不再有\_\_src，而是它本身就相当于\_\_src



## 附录

**Rsync**

特性：

* 镜像保存整个目录树和文件系统（包括文件创建时间、修改时间、权限、软硬链等）
* 效率高，速度快，可同步只修改过或新增的文件
* 安全，可使用scp、ssh等来传输，本机同步无需担心这点

使用方式：

rsync -urt --delete --exclude=pattern src dest

其中参数u设置只同步有更新的文件，r设置包含子文件夹，t设置保留源文件的创建修改时间，--delete设置若源文件不存在的文件需要删除目标文件，--exclude可指定排除指定正则匹配的文件（夹），src为源，dest为目标。

具体请参看：<http://rsync.samba.org/ftp/rsync/rsync.html>

**Inotifywait**

前提：基于inotify，需要一台2.6.13或以上的linux内核版本，可通过检查inotify是否存在来判断机子能否使用，由于inotify是采用c++编写监控事件，使用起来不太方便，需要应用一个可在shell下直接调用命令行的工具inotify-tools。Inotifywait是inotify-tools提供的实时监控并输出受影响的相关文件的功能，使用方式为：

|  |
| --- |
| inotifywait -qmre modify,create,delete,move $SRC | while read DIR EVENT FILE  do  echo $DIR $FILE $EVENT  done |

一般来说只需要监听触发的文件夹DIR、具体文件FILE和相关事件EVENT即可。

安装：

inotify-tools需要自行安装，相关地址：<https://github.com/rvoicilas/inotify-tools/wiki>

采用make&make install的安装方式可能存在调用时会报错，需要用ln -s方式把解压出来的inotify-tools文件夹下的src/inotifywait 链接到 /usr/local/bin/inotifywait，才可以在全局执行inotifywait。更多使用方式请参考上面提供的地址。