Snack构建平台

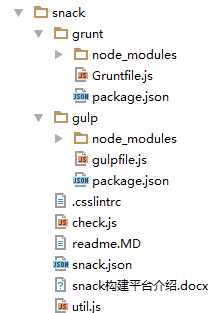
# 简介

snack是一个基于gulp和grunt构建工具编写的一个通用构建平台。

用户只需按约定，组织json数据结构，然后一个命令，就能为一个或者多个工程构建前台文件。

平台由grunt模块（暂停更新）、gulp模块（持续更新）、用户json数据（snack.json）、json数据校验模块（check.js）、通用处理模块（util.js）。

工程结构如下：



# 能力

## Less任务

该任务处理less文件。

1. 必须指定less源文件路径；
2. 支持指定输出目标路径；
3. 支持文件过滤；
4. 支持文件排除；
5. 支持合并成一个css文件；
6. 支持css文件压缩；
7. 支持对源文件路径的监听（有变化时会自动执行less任务）

## Css任务

该任务处理css文件。

1. 必须指定css源文件路径；
2. 支持指定输出目标路径；
3. 支持文件过滤；
4. 支持文件排除；
5. 支持合并成一个css文件；
6. 支持css文件压缩；
7. 支持对源文件路径的监听（有变化时会自动执行css任务）

## Js任务

该任务处理js文件。

1. 必须指定js源文件路径；
2. 支持指定输出目标路径；
3. 支持文件过滤；
4. 支持文件排除；
5. 支持合并成一个js文件；
6. 支持js文件压缩；
7. 支持对源文件路径的监听（有变化时会自动执行js任务）

## Html任务

该任务处理html文件。

1. 必须指定html源文件路径；
2. 支持指定输出目标路径；
3. 支持文件过滤；
4. 支持文件排除；
5. 支持合并成一个html文件；
6. 支持html文件压缩；
7. 支持对源文件路径的监听（有变化时会自动执行html任务）

## Copy任务

该任务处理所有文件的复制。

1. 必须指定源文件路径；
2. 支持指定输出目标路径；
3. 支持文件过滤；
4. 支持文件排除；
5. 支持对源文件路径的监听（有变化时会自动执行copy任务）

# 版本

当前版本为0.1 beta

# 准备工作

1. 安装node.js环境，window下下载exe文件安装即可；
2. 如果在gulp模块里没有node\_modules目录或者在执行命令的时候有插件丢失异常，则在gulp目录下，删除node\_modules目录，并命令行执行以下命令安装插件：npm install（grunt同理）；

# 使用指南

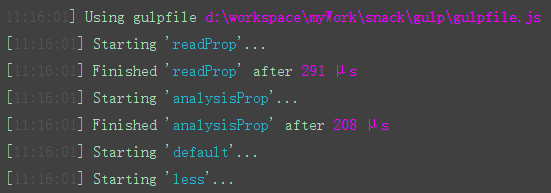
## 1、编辑用户json数据结构，即snack.json文件。

Snack.js文件例子（详细说明请参见readme.MD文件）：

{  
 **"basePath"** : **"../../"** ,//工程相对于gulpfile.js文件的相对路径  
 **"projects"** : [  
 {  
 **"name"** : **"vita"** ,//工程名  
 **"base"** : **""** ,//该工程里的基础路径，可不设置  
 **"less"** : {//less任务  
 **"watch"** : **true**,//监听总开关，可以在paths里每个节点单独设置  
 **"paths"** : [  
 {  
 **"source"** : **"themes/default"**,//文件源路径  
 **"filter"** : **"theme"**,//过滤的文件名  
 **"target"** : **"dist/themes/default"**,//目标路径  
 **"minify"** : **true**//压缩}  
 ]  
 } ,  
 **"js"** : {//js任务  
 **"watch"** : **true**,  
 **"paths"** : [  
 {  
 **"source"** : **"core"**,  
 **"filter"** : [**"core"**,**"instance"**,**"queue"**,**"event"**,**"canvas"**,**"aop"**,**"widget.factory"**,**"widget"**,**"modular"**],  
 **"target"** : **"dist/core"**,  
 **"fileName"** : **"vita-core"**,//合并的文件名  
 **"minify"** : **true** }  
 ]  
 } ,  
 **"copy"** : {//复制任务  
 **"watch"** : **true**,  
 **"paths"** : [  
 {  
 **"source"** : **"themes/fonts"**,  
 **"target"** : **"dist/themes/fonts"** }  
 ]  
 }  
 }  
 ]  
}

## 2、进入gulp目录，命令行执行gulp命令

就可以看到如下输出



## 3、注意：关于watch（监听）

只要有设置任何一个watch，则该进程就会一直在等待，监听着指定目录，在目录内容发生变化的时候，会自动执行对应任务。如果关闭进程，那么将会停止监听，切记。