# 编译工具之Build-Task

By pakinguo

## 工具开发原因

目前Web前端架构采用基于CMD的Seajs模块加载引擎，使用模块系统对于前端来说大大提高开发效率、降低模块和页面间的偶尔度、防止作用域污染等等。使用Seajs开发起来，开发者只需关注功能和业务，不需要关注部署和发布。这时，问题来了，开发中肯定很多模块各种引用加载，其中某些模块是可以合并压缩，或者，某些静态资源（包括图片和css）可以做更多优化，等等。性能优化需要整合，但如果有自动化的工具来构建项目，节省很多时间，同时也达到优化的目的。

对于CMD的Seajs，已经有很健全的项目构建工具可以进行自动化和部署，但是，由于是第三方构建工具，其构建出来的项目文件结构并不适用于我们本身的项目（特别是国际化），加上一些细节上的处理与预期不一致，如期望在开发过程，开发可以即改即浏览，而构建出来的项目文件目录等也只通过简单的配置达到与开发时看到的页面效果一摸一样，而不需要去改变太多的路径索引（这里只包括将html片段转化为Seajs模块，即使Seajs提供了seajs-text插件支持异步请求html或tpl或json文件，但构建后依然是异步请求，并没有合并到js模块中）。还有Seajs新版本移除了对css的require支持，将其作为一个插件由开发者自行决定是否使用，而Seajs同时提供了seajs-style可以内联编译好的css文件（转换为seajs模块），这两者如何不需要改变任何require的模块路径来达到快速切换？另外，Seajs提供了seajs-combo来对请求的资源进行合并（需要后台支持），这样需要php或者某台服务器配置好环境，请求的时候做合并（若有缓存直接读缓存文件），无论如何，第一次请求肯定需要花费一定的时间，是否可以在部署前就完成静态资源的合并，根据请求规则直接加载编译好的文件呢（由于上线后的项目只会定期发布新版本去update模块、业务功能，故大部分代码，排除第三方框架，基本上都不会有改动）？还有……

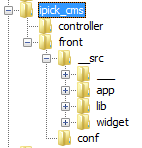
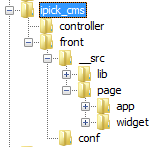
鉴于以上的问题，需要在提供的构建工具基础上，做一部分功能的适配，来适应自己项目的构建。

## 工具功能

1. 对指定的项目进行cmd-transport，将开发的模块转化为seajs完整模块（带id、依赖，这样运行代码的时候seajs就不需要额外去解析该模块）
2. Transport后，根据项目需要，生成自动合并的文件列表，通过contrib-uglify对合并的文件进行混淆压缩

## 编译流程

### 文件目录规范（为适用于国际化）



其中，front（可命名为resource）与php的controller同级；

\_\_src与conf同级，\_\_src为开发时的目录，里面包括静态资源和页面、模板资源，发布时不会同步此文件夹；conf里有页面需要引用的配置文件、配置文件的模板文件、seajs模块的alias列表文件、构建时需要忽略合并的alias列表；

lib是第三方框架放置的地方，其不参与构建，只参与文件同步；

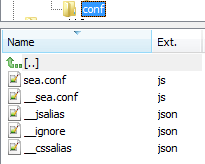
page（或者同级直接为app和widget等）与lib同级，放置页面业务模块或组件模块。

### 项目开发过程

在开发机上会部署同步文件的功能（synchronize-files工具），通过编写一个Build-realtime的脚本，配合crontab做定时同步任务。主要是将开发目录下的文件同步一份到开发环境运行的真实目录下，此时不做任何构建，只复制生成配置文件和同步开发文件。

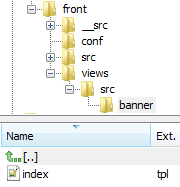
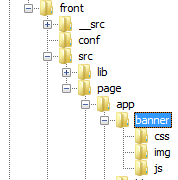
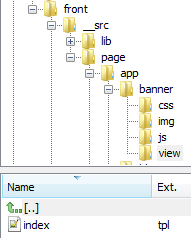
Build-realtime主要做以下的功能：

* 生成seajs的前端配置文件
  + 相关组件：【alias-conf】
  + 需要参数：配置文件的模板文件路径（一般放在conf下，如下图）、seajs模块中js的别名alias列表、seacss模块中css的别名alias列表、输出的文件路径



* 同步目录及文件
  + 相关组件：【synchronize-files】
  + 提供两个方法：compareExclude和compareOnly；compareExclude是用来复制目录下文件夹中除了文件夹名为view的所有文件，保持源目录结构和复制后的一致，这里主要是同步非页面模板（php来解析的smarty模块.tpl文件），默认指定为view的文件夹；compareOnly则是将view文件夹的模板文件复制到指定的路径下。（如下图示例）

=>



开发在\_\_src里面编写代码，由实时同步工具生成到src和views里，这样，所有tpl、js和css都在一个文件夹下编写，减少了开发中因为模块过多而导致tpl与js、css来回查找的时间，提高效率。同时，以模块名作为文件夹的名称，就不用再对css和js做命名一致性，只需要保证文件夹命名就行。

\*备注：遗留一个将.htpl自定义的html片段模板编译为CMD模块的功能没完成。

### 编译部署

开发完成后，要发布到测试环境，需要对代码文件进行合并压缩，只需要运行编写好的脚本：build-task.js。Build-task组成的组件有：

* 【cmd-transport】：对模块进行cmd-transport，由于这里配合使用gruntjs的构建工具，需要两个步骤来完成：先生成Gruntfile.js，再执行grunt进行构建。
* 【cmd-uglify】：transport之后，以文件为单一模块，文件所依赖到的模块进行合并压缩，其中排除需要忽略合并的模块，通过conf下的\_\_ignore.json来指定（文件格式与\_\_jsalias.json一致）。
* 其他依赖的文件或参数：

CONFIG.json：构建读取的配置文件，包含所有需要参数的文件，可有build-task下的【build-init】工具按照提示自己指定，或者修改build-task下的配置文件，参数如下列表：

|  |  |
| --- | --- |
| TRANSPORTSRC | Transport的源路径，如”D:/ipick\_cms/front/src” |
| TRANSPORTDEST | Transport输出的路径，如"D:/ipick\_cms/front/en-us" |
| UGLIFYSRC | Uglify指定的源路径，如”D:/ipick\_cms/front/en-us/page” |
| UGLIFYBASE | Uglify合并文件除了Transport的相对路径，还需要根目录的路径，即是base路径，一般与TRANSPORTDEST取值一致 |
| ALIAS | 开发中引用到的Seajs的alias |
| IGNORE | 编译需要忽略合并压缩的模块alias |
| GRUNTJS | 默认路径为”../gruntjs”，一般不需要修改，为gruntjs构建工具的目录 |

## 附：编译工具所有组件

|  |  |
| --- | --- |
| alias-conf | 生成seajs的配置文件，用于前端页面的配置模块规则等 |
| synchronize-files | 同步开发目录下的文件，按规则输出 |
| cmd-transport | Transport指定文件目录 |
| cmd-uglify | Uglify transport后的文件 |
| gruntjs | Gruntjs构建工具，辅助cmd-transport和cmd-uglify |
| util-mkdir | 同mkdir |
| util-mv | 同mv |
| util-rm | 同rm |
| build-realtime | 实时文件同步，主要执行alias-conf和synchronize-files |
| build-task | 编译项目 |