

前端workflow

情封 @UED

2014.07.29

主要内容

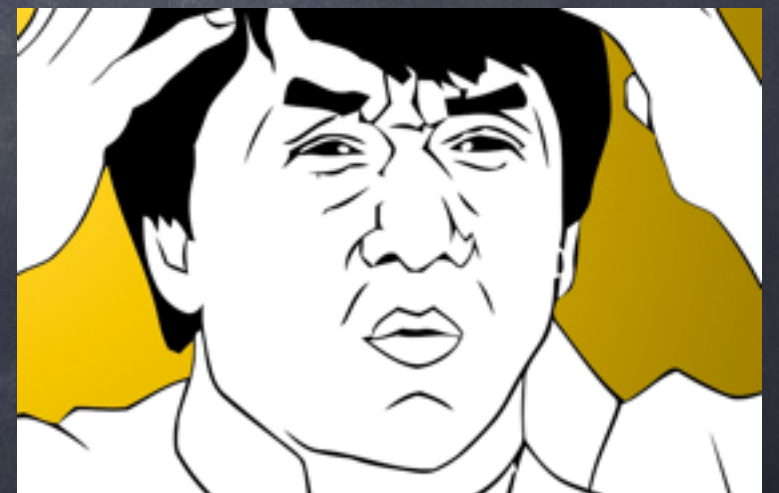
- 回顾
- 构建、脚手架
- 案例：手游平台

初印象

那些年

- 游戏官网、专题、活动页、平台型页面
- 前后端交互js、js组件
- 套php模版
- 玩点php

做的最多的都是写静态页



下面的场景，你体验过吗？

场景一：项目开始前

- 先建立一个 projA 的文件夹 再建 html css js images （建5个或以上文件夹，花费1分钟）
- 拷贝 CSS库（Yui Reset | bootstrap）JS库（Requiet.js | Seajs | jQuery | jQuery插件）进相应目录（拷贝 N个文件，花费N分钟）
- 再新建 html/index.html js/comm.js css/base.css css/comm.css css/module-A.css （建N个文件，花费N分钟）

场景二：编码ing

- 编辑器编码 => 切换到浏览器F5 => 编辑器编码
=> 切换到浏览器F5 => 编辑器编码 => 切换到
浏览器F5 => 编辑器编码 => 切换到浏览器F5

.....

不要在意这些细节

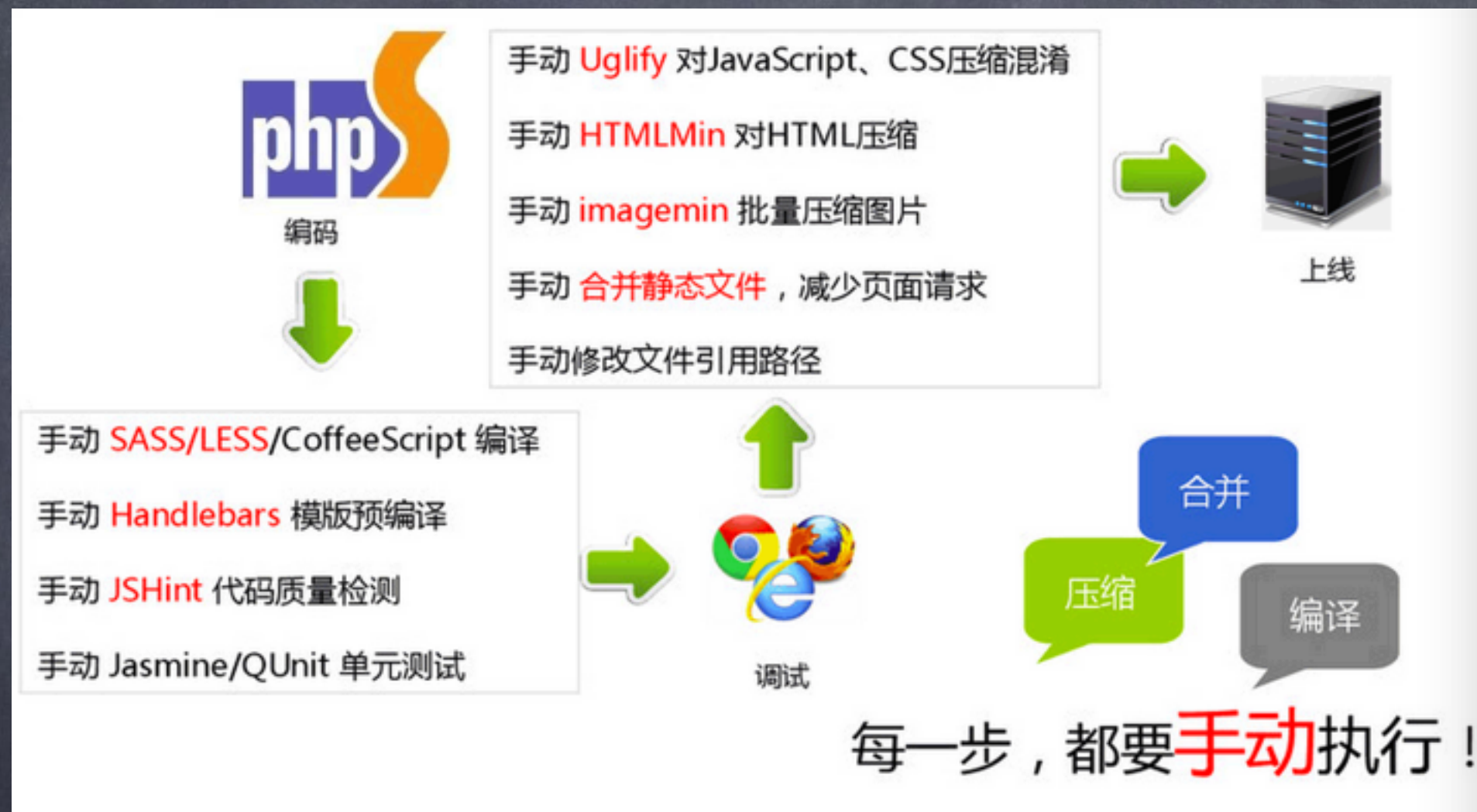


场景三：编码完成

- HTML去掉注释、换行符 - HtmlMin
- CSS文件压缩合并 - CssMinify
- JS代码风格检查 - Jshint
- JS代码混淆压缩 - Uglyfy
- image压缩 - imagemin

发布前，最后一件事：

```
$ java -jar yuicompressor-x.y.z.jar myfile.js -o myfile-min.js --charset utf-8
```

传统开发流

如此的简单，每次都在重复。

你还爱吗？

救世主

Gulp

Grunt

Grunt介绍

<http://www.gruntjs.com>

Grunt介绍

- 基于nodeJS的命令行工具，可用于自动化构建，测试，生成文档的管理工具。
- 一个自动任务运行器，会按照预先设定的顺序自动运行一系列的任务



Grunt介绍

- 编译文档型：less , sass , coffeescript等
- 文件操作型：合并、压缩css , js等
- 质量保障型：mocha , jshint , csslint等
- 类库构建型：ember.js,backbone.js等

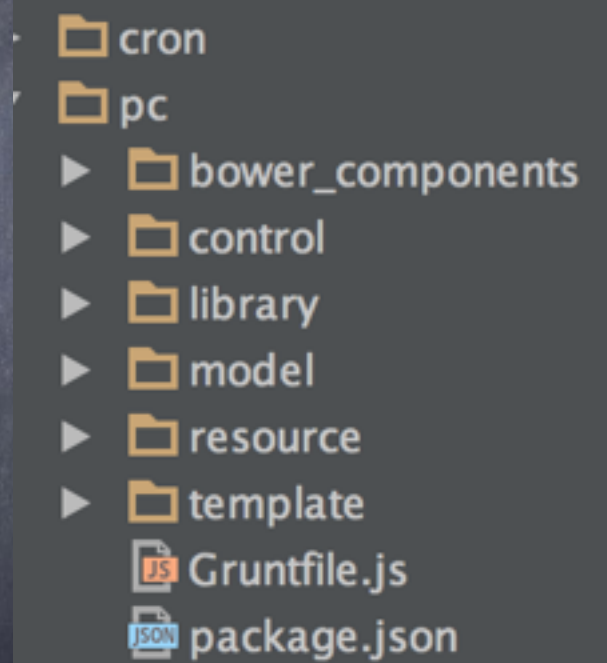
Grunt介绍

- Gruntfile.js

项目自动化工作流程配置文件

- package.json

项目信息，模块依赖声明



Gruntfile主要有三个方法：

1. `grunt.initConfig` // 定义模块的参数配置
2. `grunt.loadNpmTasks` // 引用声明，完成任务所需的模块加载
3. `grunt.registerTask` // 定义具体的任务


```

* */
var specialLess = [...];

grunt.initConfig({
  pkg : grunt.file.readJSON('package.json'),
  less : {
    android: {...},
    special: {...},
    feedback: {...}
  },
  concat: {...},
  cssmin : {...},
  template: {...},
  copy: {...},
  clean: ['resource/dev/css/temp/'],
  watch:{...}
});

grunt.loadNpmTasks('grunt-contrib-less');
grunt.loadNpmTasks('grunt-contrib-cssmin');
grunt.loadNpmTasks('grunt-contrib-imagemin');
grunt.loadNpmTasks('grunt-contrib-concat');
grunt.loadNpmTasks('grunt-contrib-copy');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.loadNpmTasks('grunt-template');
grunt.loadNpmTasks('grunt-contrib-clean');
//grunt.registerTask('android', ['less:android', 'concat:android']);

/*...*/
grunt.registerTask('android', ['less:android', 'concat:android']);

/*...*/
grunt.registerTask('special', ['less:special', 'concat:special']);

/*...*/
grunt.registerTask('feedback', ['less:feedback', 'concat:feedback']);

```



```
{
  "name": "4399cn",
  "version": "0.1.0",
  "devDependencies": {
    "grunt": "~0.4.1",
    "grunt-contrib-uglify": "~0.2.2",
    "grunt-contrib-cssmin": "~0.6.1",
    "grunt-contrib-concat": "~0.3.0",
    "grunt-contrib-watch": "~0.4.4",
    "bower": "~1.2.7",
    "grunt-inline": "~0.2.0",
    "grunt-cmd-transport": "~0.4.1",
    "grunt-cmd-concat": "~0.2.7",
    "grunt-htmlhint": "~0.4.0",
    "grunt-contrib-less": "~0.9.0",
    "grunt-contrib-copy": "~0.5.0",
    "grunt-template": "~0.2.3",
    "extend": "~1.2.1"
  },
  "dependencies": {
    "time-grunt": "~0.2.3"
  }
}
```




前端自动化构建

优点

- 跨平台、易维护、易迁移、开发简单
- 符合前端发展趋势
- 配置文件灵活，可利用第三方插件，也可自行扩展

善于利用工具，让时间去做有意思的事情。

缺点

- 依赖nodejs环境，在window环境下体验不是很好。如图片优化插件 `grunt-contrib-imagemin`
- 每次启动项目，需手动在命令行中运行

项目脚手架

项目构建

web项目构建

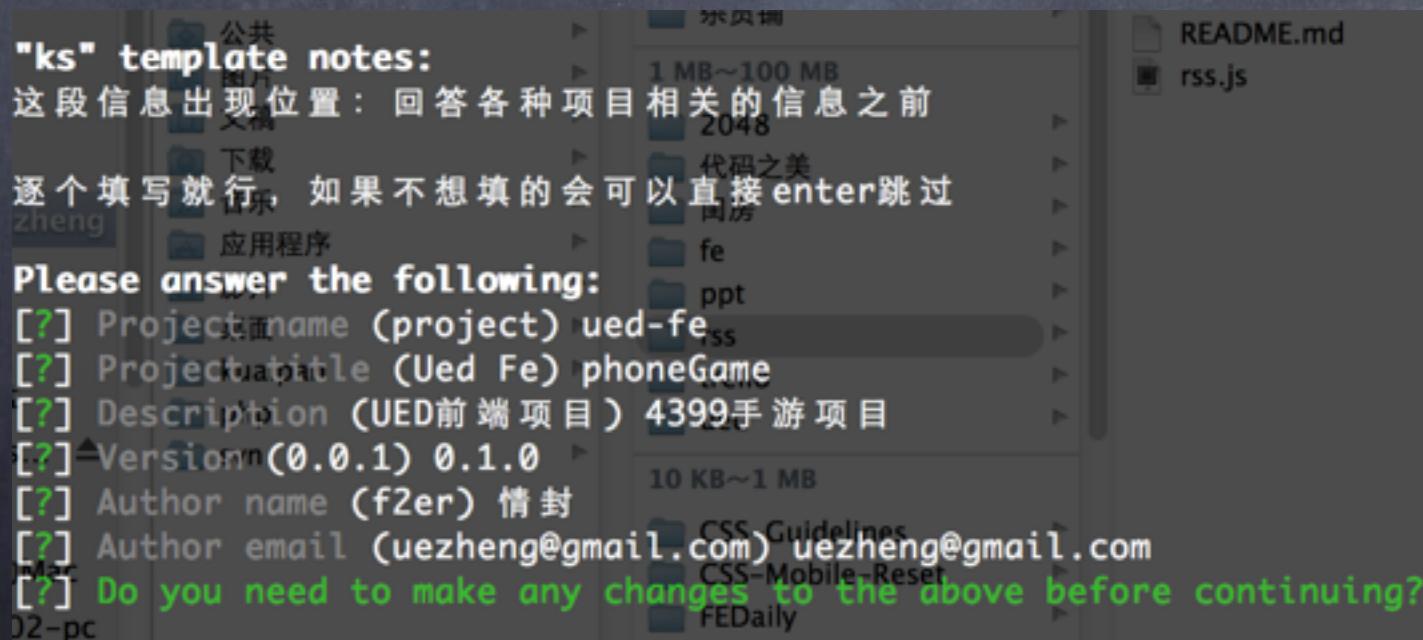


mobile项目构建



KS

- 基于现在的团队情况 本地项目文件初始化
- 打包通用的插件 合并、压缩、代码编译
- 静态资源的管理 bower



KS使用

安装grunt相关的插件，然后两行命令

```
grunt install
```

```
bower install
```


案例

- 手游平台
- 页游
- 游戏收藏盒
- . . .

案例：手游平台

- 开发版本、发布版本

- js、css模块化

js模块加载器：seajs

css预处理器：less

包管理器：bower

- 静态资源缓存问题

时间戳、文件重命名、新建目录、hash

谢谢