



—— 智慧企业领导者 ——

前端工作规范
V1.0

迈动互联（北京）信息科技有限公司
2016.07.31

目 录

- 一 环境搭建及项目模板检出
- 二 目录结构（PC/WAP），依赖安装
 - 1. Html文件的命名；
 - 2. img文件的存放位置；
 - 3. js文件的存放位置；
 - 4. less文件存放位置；
 - 5. 第三方库的存放位置；
- 三 模板使用
 - 1.指令启动；
 - 2.文件解释；
 - 3.svn提交；
- 四 个人周报

一 环境搭建

因我们的站点是基于gulp构建的，所以我们在使用gulp进行构建时,需要安装所需要的软件nodejs、Git
版本控制我们使用TortoiseSVN

下载地址：

nodejs: <http://nodejs.cn/>

git: <http://rj.baidu.com/soft/detail/30195.html?ald>

TortoiseSVN: <http://pan.baidu.com/s/1eSx6hwx>

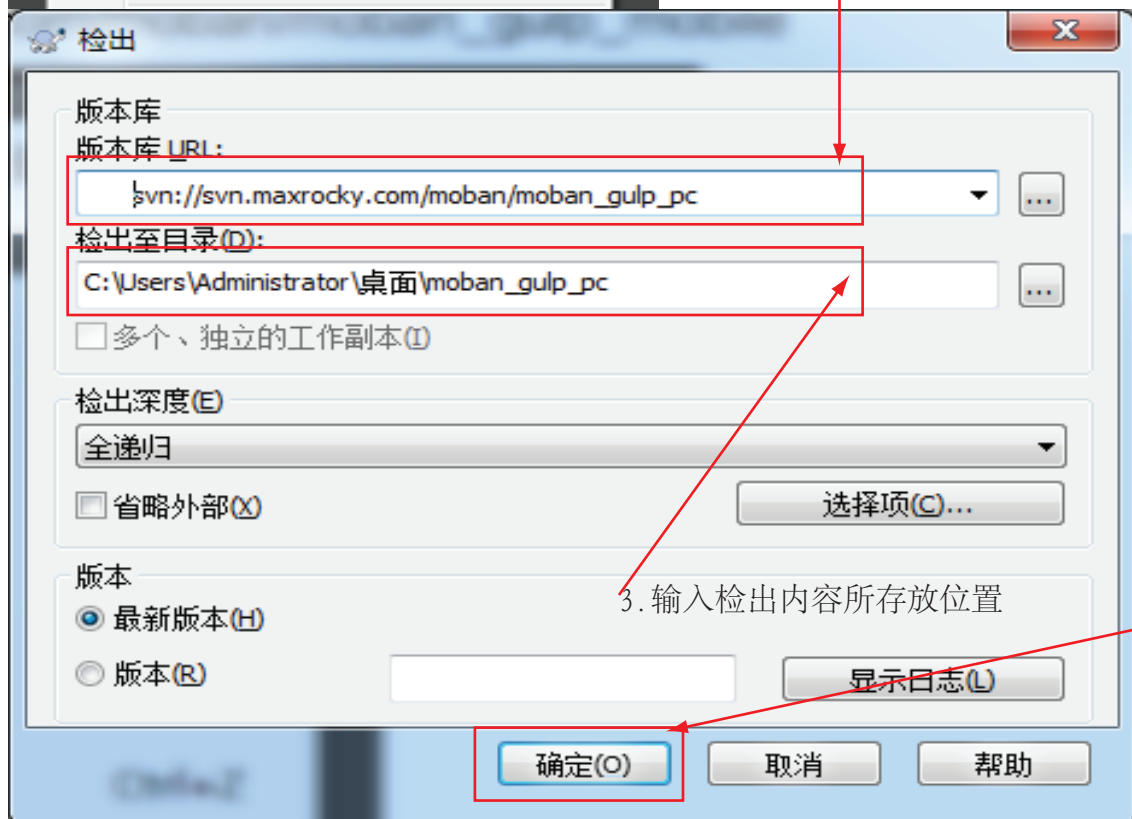
个人可根据系统位数（64位/32位）进行安装，首次将三个软件安装完毕后，可向我索要svn帐号及密码(用户名：名字全拼、密码：随机)

软件安装不在此处阐述....



1. 右键呼出菜单

2. 输入检出地址



3. 输入检出内容所存放位置

4. 确定

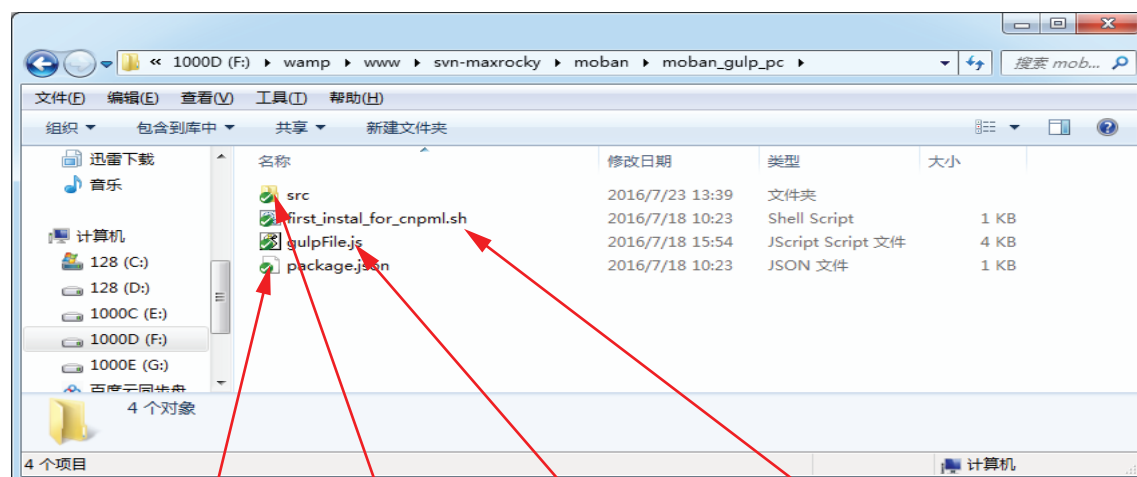
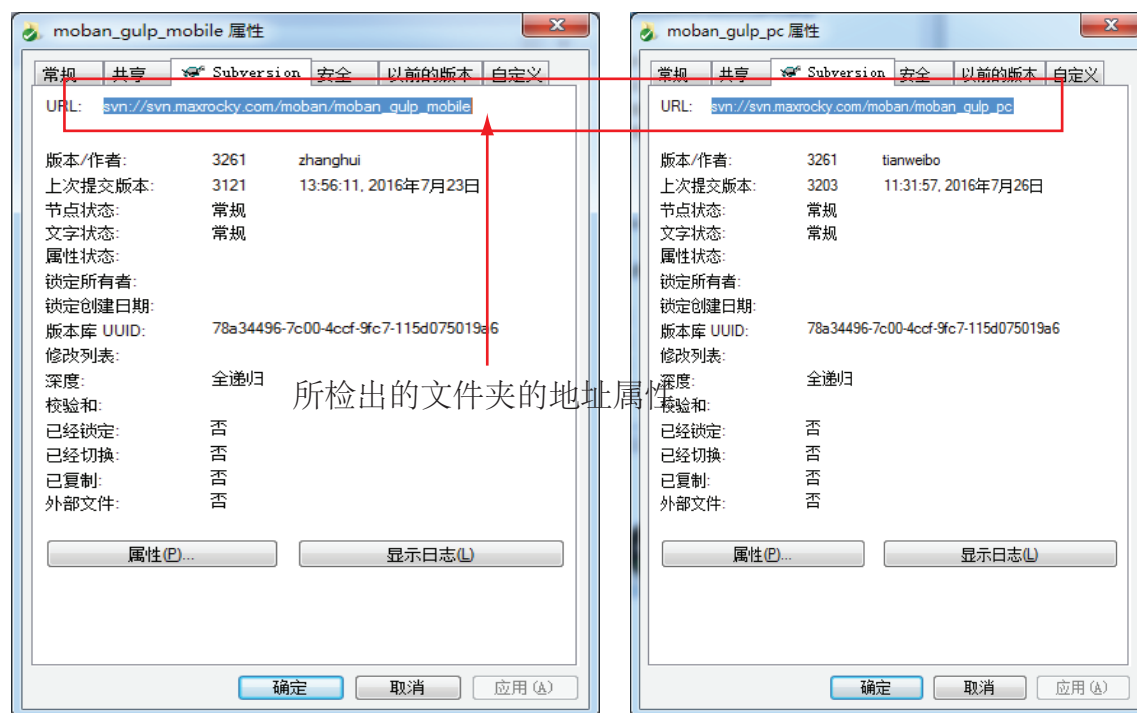
二 目录结构

上述软件安装完毕后，请将svn地址：

svn://svn.maxrocky.com/moban/moban_gulp_mobile

svn://svn.maxrocky.com/moban/moban_gulp_pc

分别检出；得到如下目录及结构：



基于
nodejs的
依赖包配置
项文件！

src 文件夹
为项目源目
录，所有的
编辑在这里
进行！

基于gulp的
配置文件！

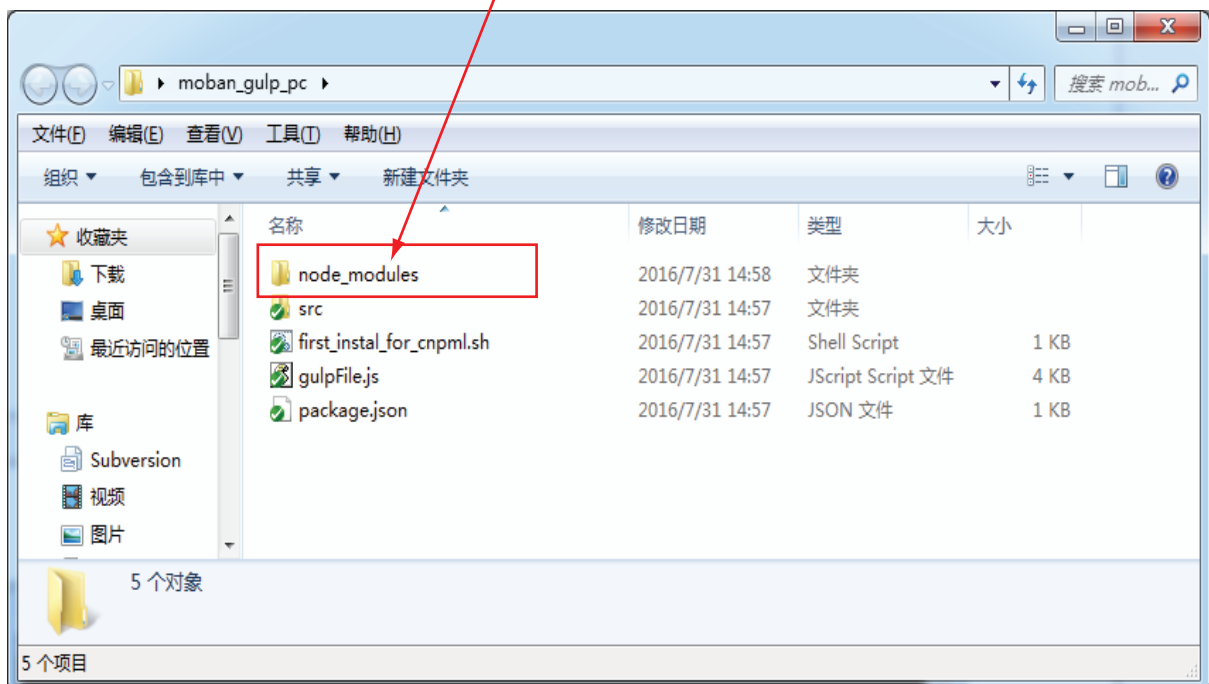
初始安装所
有依赖的安
装文件！

双击first_instal_for_cnpml.sh,进行首次安装! 界面截图如下：

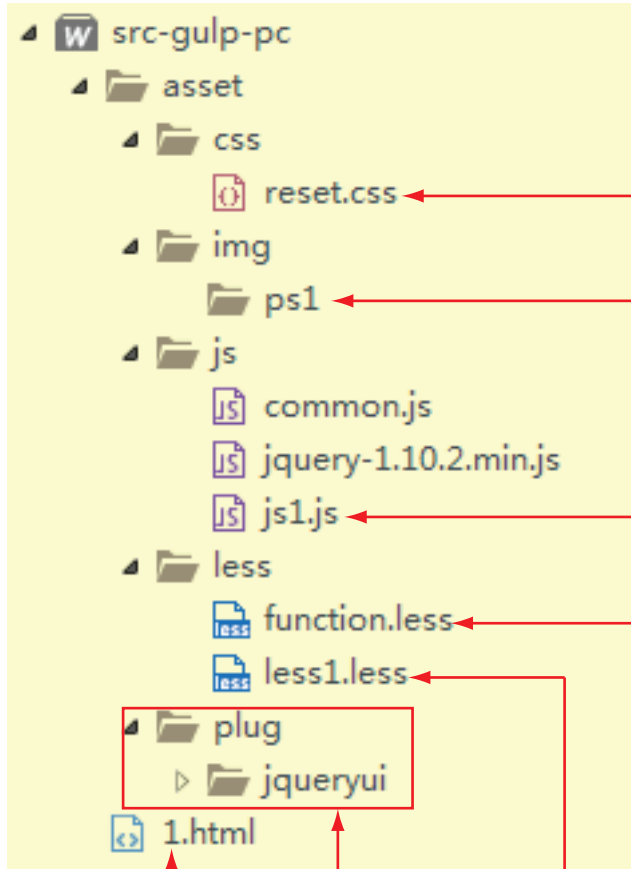
```
npm
Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.
npm WARN deprecated has-color@0.1.7: Renamed to supports-color. If you're using
chalk, upgrade to the latest version. https://github.com/chalk/supports-color
C:\Users\Administrator\AppData\Roaming\npm\cnpm-sync -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-sync
C:\Users\Administrator\AppData\Roaming\npm\cnpm-user -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-user
C:\Users\Administrator\AppData\Roaming\npm\cnpm-web -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-web
C:\Users\Administrator\AppData\Roaming\npm\cnpm-check -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-check
C:\Users\Administrator\AppData\Roaming\npm\cnpm-search -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-search
C:\Users\Administrator\AppData\Roaming\npm\cnpm -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm
C:\Users\Administrator\AppData\Roaming\npm\cnpm-doc -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm-doc
```

安装完成后生成如下标红文件夹,不用理会这个文件夹即可,但一定要有;



下面我们项目源文件(src)的目录结构:



样式重置, 实际工作中不用理会

第一位前端人员的图片目录, 所有第一位前端人员的图片请放到这个文件内!

第一位前端人员的js文件编辑位置, 如非必要将js写到页面内的话, 请将js写到这个文件里!

我们公共且常用的less库文件, 里面有一些写法, 熟悉后, 可大大加快开发速度!

第一位前端人员的样式编辑文件, 你所有的样式要在这里面进行编写, 最好是引用function.less的写法!

plug文件夹用于存放第三方支持的库或组件, 当前默认放置了一个jqueryui-datepicker的组件, 你也可以引用其它第三方库(轮播图, 视频组件, swiper)等等, 引用的第三方组件库, 请自行新建文件夹进行放置, 允许js/img/html放在自定义文件夹内!

需要我们编辑的html文件, 所有我们编辑的html文件都以”数字.html”进行命名, 即1.html, 2.html, 3.html, 100.html, 101.html, 请不要以0开头的html命名。在common.js中已经加入快速翻页功能, 此翻页功能不支持非数字的翻页。假设我们编辑了10个页面(10.html), 我们可以通过键盘“左”、“右”, 来快速对所编写的页面进行查看! 即方便了我们, 又方便开发快速定位页面。

对于一个20页以内的普通网站，我们就采用上面的作法来解决，即第一位前端来执行页面的制作，当一个站内文件超过20页的页面时，可能会采用多人协作的形式来对同一站点进行开发，这时我们就要采用不同的工作目录和文件来区分，且互不影响还能保证代码的合并，还能减少http的请求数，达到提高性能的目地！

如：第二位前端开发人员也参与到第一个前端人员的开发中，两个人要同时对一个站点进行开发，这个时候我们要对html,js,css,img进行区分：

常规的作法是：

html:

前面说过我们采用的” 数字.html” 的形式来命名，第二位开发人员可以从100.html开始进行页面的制作；第三位可以从200.html进行制作！

img:

请自行在asset/img/下新建文件夹ps2,所以第二位前面开发的图片放置在这个文件夹内；第三个新建ps3;

js:

请自行在asset/js/下新建js2.js文件，且文件名必须叫这个文件，同理第三个前页来参与时的文件为js3.js;

css:

因我们全部样式都采用less进行编写，所以请在asset/less/中新建less2.less,第三人请新建less3.less

这样每一个开发人员各自编辑自己的html,css,js,img文件，互不干扰。有一点需要注意的是：当第二位甚至第三位参与且同时进行开发时，请第二位和第三位对自己所编辑的样式名，进行有效的命名，如第一位开发命名了一个.title{}样式，第二位开发时最有效的命名.zh_title{},即我的名字的首字母_类名进行样式名的编写，这样可以有效且大大降低多人协作时，合并代码样式时出现的互相覆盖的问题！

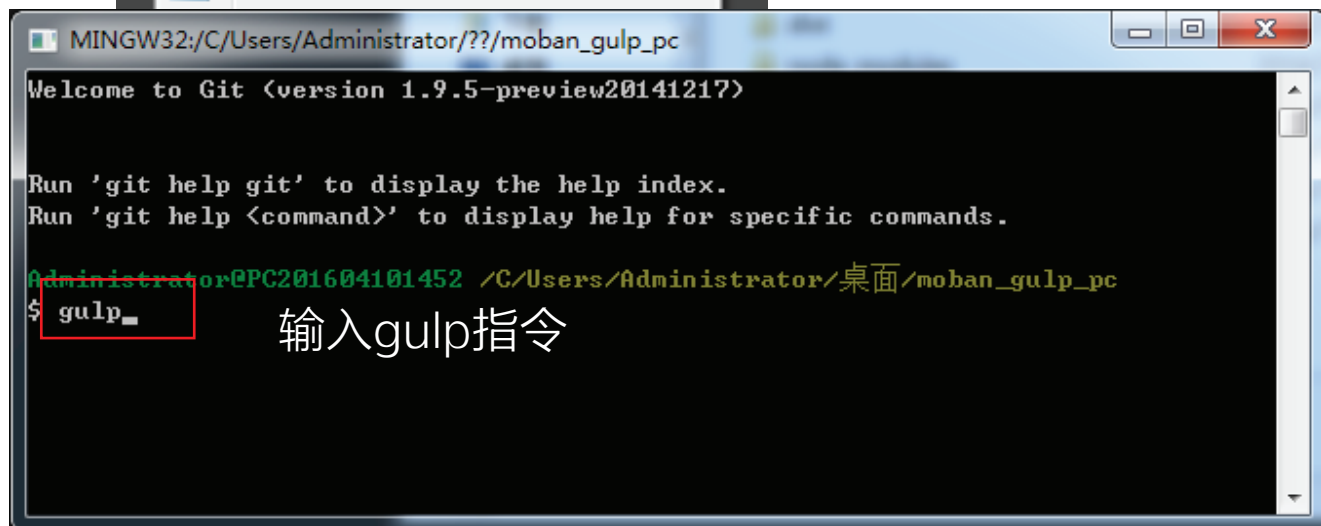
三 模板使用

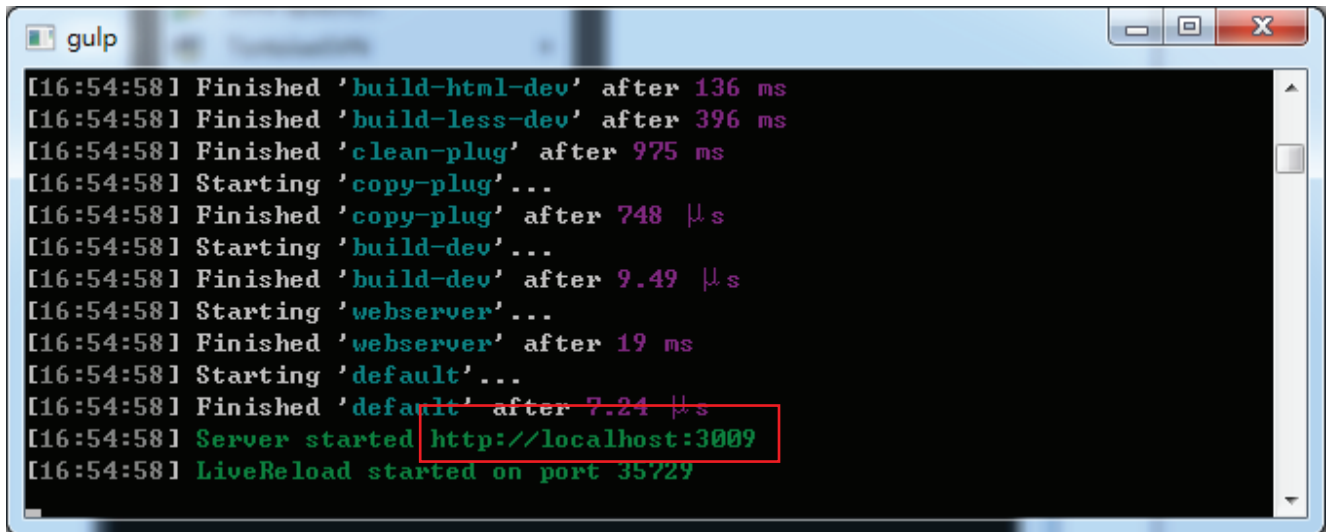
我们的模板分别两套，一套应用于pc，一套应用于wap，针对不同的制作需要可分别使用不同的模板进行开发工作；

首次使用时，请在当前文件夹内右键，选择



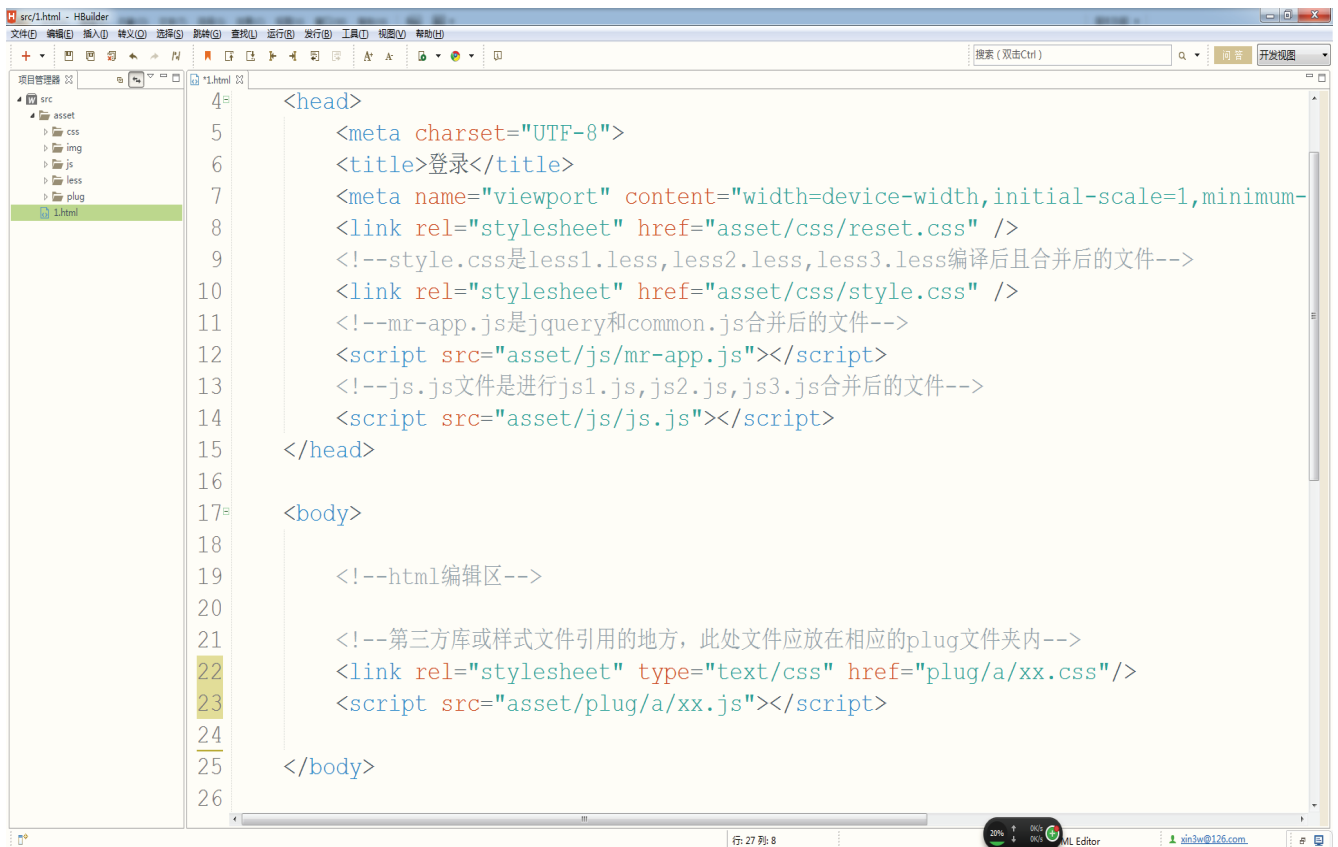
第一步



A terminal window titled 'gulp' showing the output of a build process. The output includes timestamps and durations for various tasks: 'build-html-dev' (136 ms), 'build-less-dev' (396 ms), 'clean-plug' (975 ms), 'copy-plug' (748 μs), 'build-dev' (9.49 μs), 'webserver' (19 ms), and 'default' (7.24 μs). The final line indicates the server is started at 'http://localhost:3009' and LiveReload is on port 35729. A red box highlights the 'Server started' line.

```
[16:54:58] Finished 'build-html-dev' after 136 ms
[16:54:58] Finished 'build-less-dev' after 396 ms
[16:54:58] Finished 'clean-plug' after 975 ms
[16:54:58] Starting 'copy-plug'...
[16:54:58] Finished 'copy-plug' after 748 μs
[16:54:58] Starting 'build-dev'...
[16:54:58] Finished 'build-dev' after 9.49 μs
[16:54:58] Starting 'webserver'...
[16:54:58] Finished 'webserver' after 19 ms
[16:54:58] Starting 'default'...
[16:54:58] Finished 'default' after 7.24 μs
[16:54:58] Server started http://localhost:3009
[16:54:58] LiveReload started on port 35729
```

浏览器内输入localhost:3009即可打开目录，对html文件进行预览

A screenshot of the HBuilder IDE showing an HTML file named '1.html'. The file contains a head section with meta tags for charset, viewport, and titles, and a body section with comments and links to CSS and JS files. The project structure on the left shows folders for 'src', 'asset', 'css', 'img', 'js', 'less', 'plug', and '1.html'. The status bar at the bottom shows '行: 27 列: 8' and 'ML Editor'.

```
4 <head>
5   <meta charset="UTF-8">
6   <title>登录</title>
7   <meta name="viewport" content="width=device-width,initial-scale=1,minimum-
8   <link rel="stylesheet" href="asset/css/reset.css" />
9   <!--style.css是less1.less,less2.less,less3.less编译后且合并后的文件-->
10  <link rel="stylesheet" href="asset/css/style.css" />
11  <!--mr-app.js是jquery和common.js合并后的文件-->
12  <script src="asset/js/mr-app.js"></script>
13  <!--js.js文件是进行js1.js,js2.js,js3.js合并后的文件-->
14  <script src="asset/js/js.js"></script>
15 </head>
16
17 <body>
18
19   <!--html编辑区-->
20
21   <!--第三方库或样式文件引用的地方，此处文件应放在相应的plug文件夹内-->
22   <link rel="stylesheet" type="text/css" href="plug/a/xx.css"/>
23   <script src="asset/plug/a/xx.js"></script>
24
25 </body>
26
```

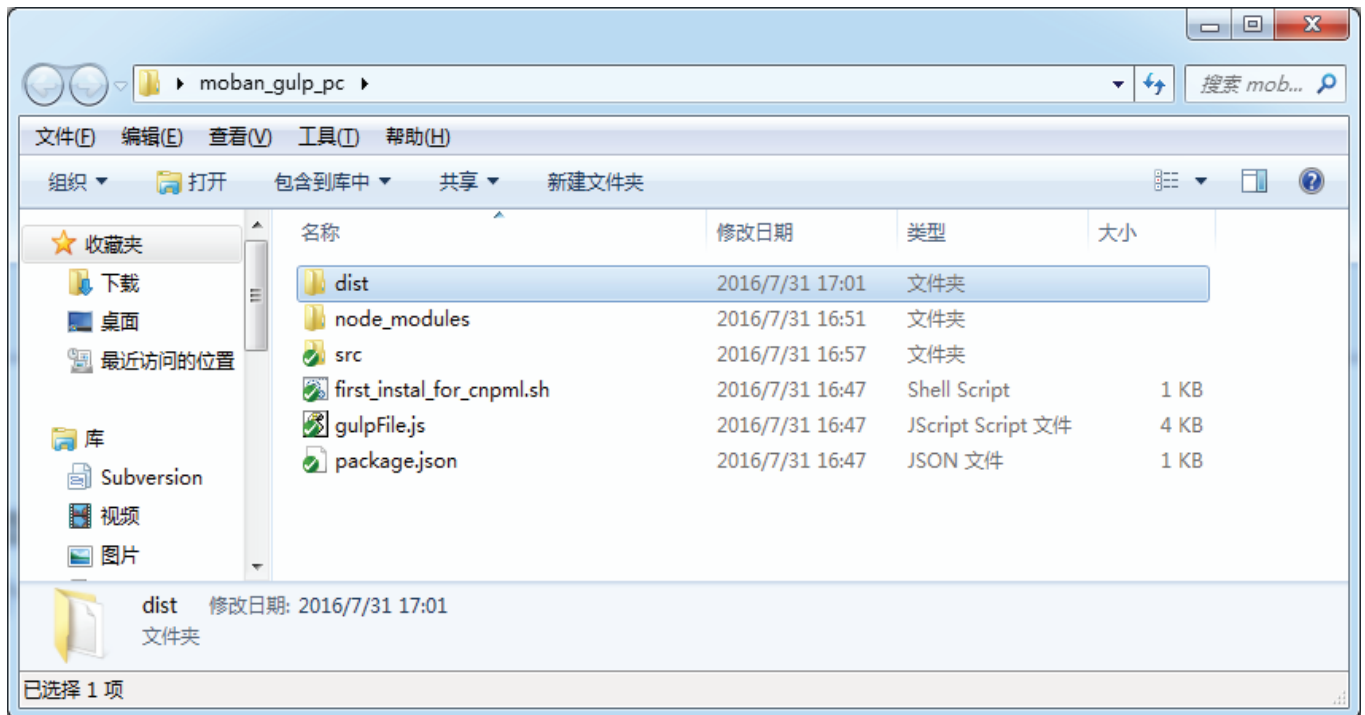
上图为html引用文件的描述：

<!--style.css是less1.less,less2.less,less3.less编译后且合并后的文件-->

<!--mr-app.js是jquery和common.js合并后的文件-->

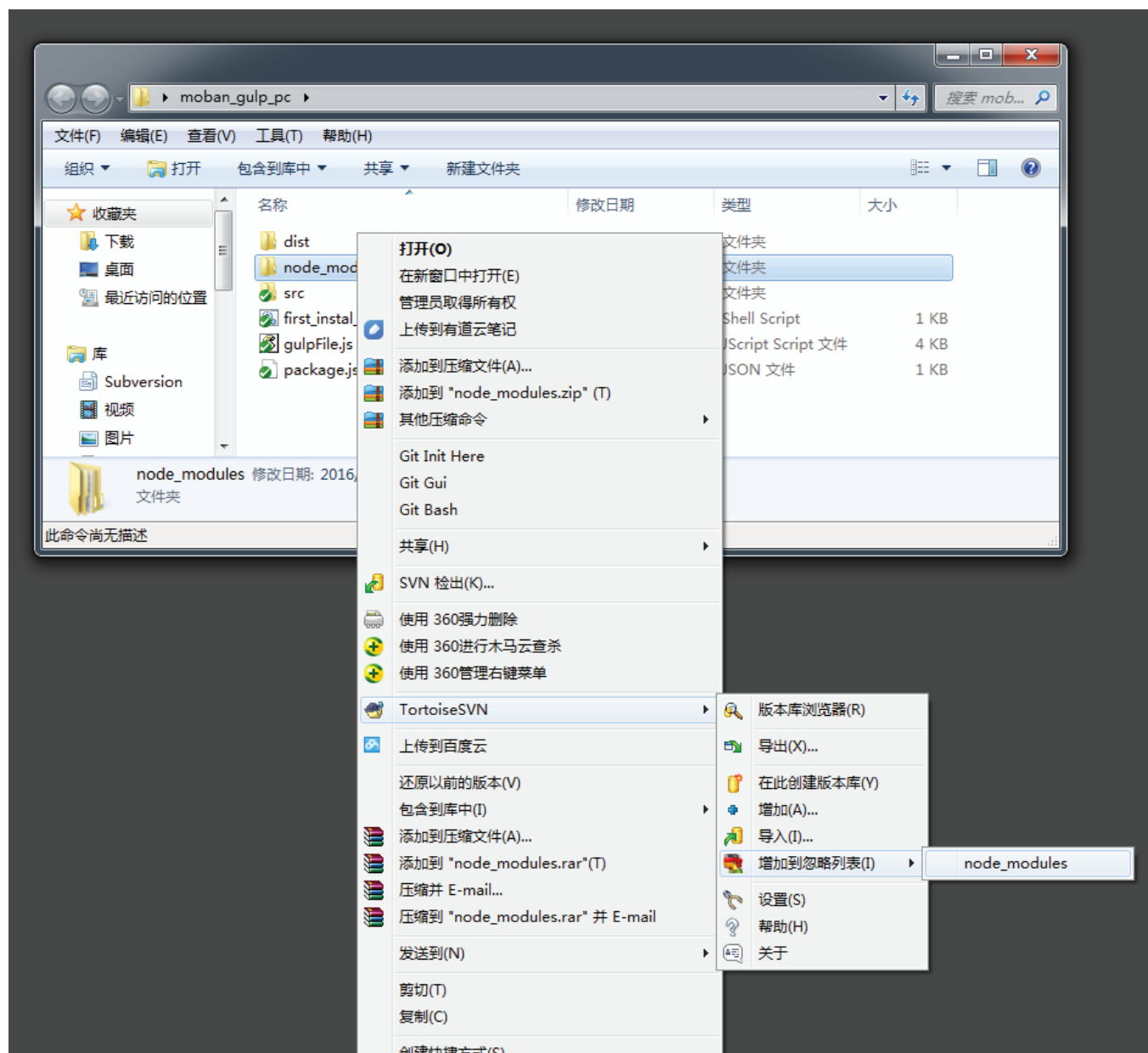
<!--js.js文件是进行js1.js,js2.js,js3.js合并后的文件-->

这时在当前项目文件夹内会生成dist文件夹，这个文件夹是就我们编辑src时自动生成的项目，这个项目是用于后端开发人员所用的文件夹!也就是说当我们src编辑完成后，所生成的文件夹正是后端所用的！也是我们所提供的！

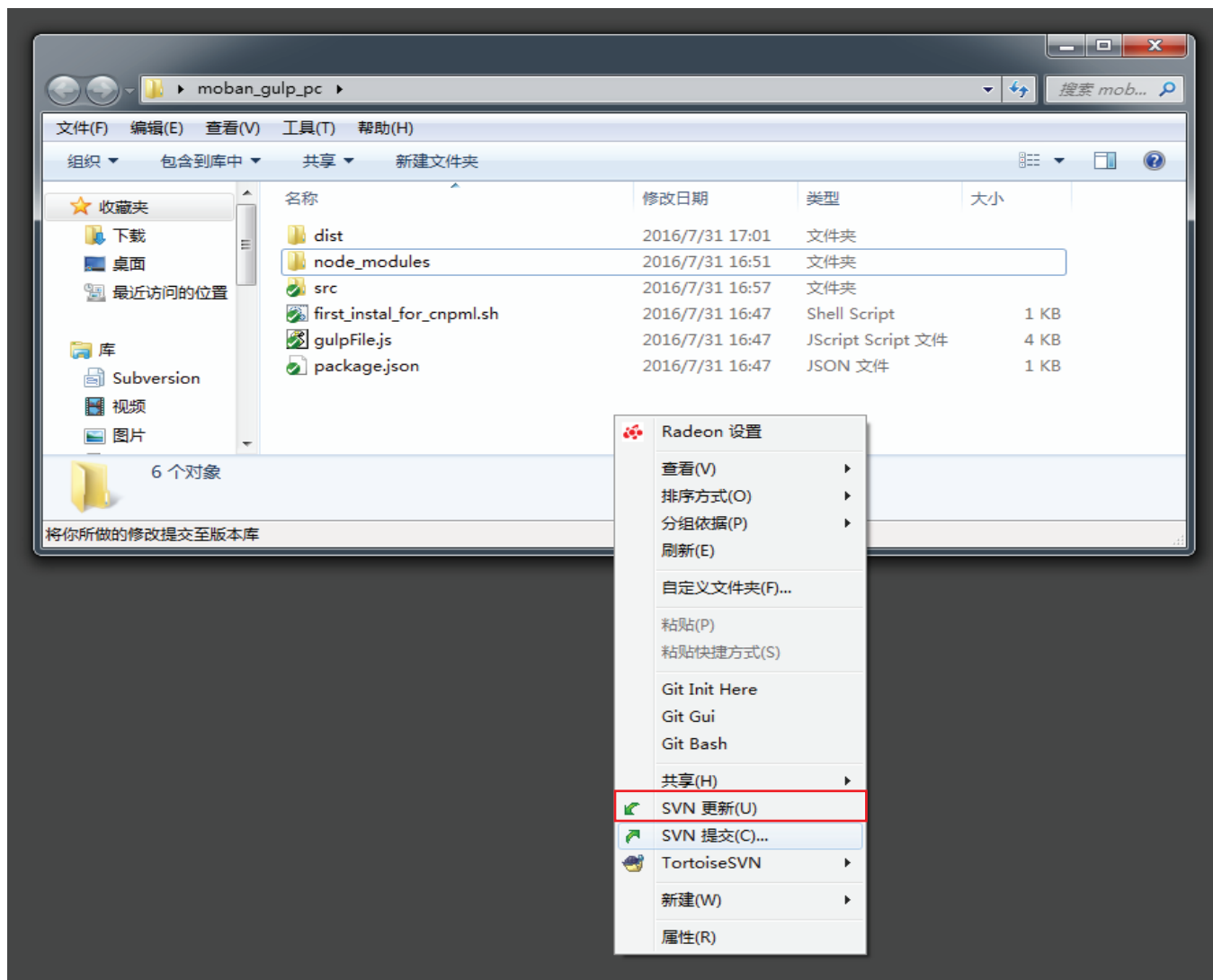


最后一步就是项目的提交：

1.提交前请先按照下图，将node_modules文件夹加入到忽略列表里，也就是这个文件夹不在我们提交的范围！



2.切记提交前要先更新，即下面标红的位置！



3.更新完成后，我们在点击svn提交！至此我们的这个流程基本算是完毕了！

四 个人周报

每周五下班前，通过邮件的方式发送本周个人周报。

格式及核心内容如下（直接在邮件中写）：

邮件标题：XXXX工作周报

邮件正文样例：

本周工作总结及下周工作安排，如下：

一：本周工作总结

1：事项1已XXXXXXXXXXXXXXXXXXXXX

2：事项2XXXXXXXXXXXXXXXXXXXXX

二：下周工作安排

1：XXXXXXXXXXXXXXXXXXXXX

2：XXXXXXXXXXXXXXXXXXXXX

三：其他

1：XXX项目问题。