## Package.json文件个字段描述

每一个项目下都有一个package.json文件

这个文件定义了这个项目所需的模块，以及各项目的配置信息，可以使用npm install命令根据这个配置文件自动下载所需模块最简单的参数如下：

name -项目名称 version -版本

### scripts字段(“scripts”:{})

Scripts 指定了运行脚本命令的npm命令的缩写，比如:

“scripts”:{

“start”:”node server.js”,

}

可以使用 npm run start代替执行node server.js命令

### dependencies字段,devDependencies字段

dependencies(”dependencies”:{}) 指定了项目运行所依赖的模块

devDependencies(“devDependencies”:{})指定项目开发所以要的模块

package.json文件可以手工编写，也可以使用npm init -y命令自动生成

如果一个模块不在package.json文件之中，可以单独安装这个模块如：

npm install express --save

npm install express --save-dev (ps:express是一个node.js的web框架)

上面代码代表单独安装express模块，--save参数表示将该模块写入dependencies属性中，--save-dev表示将该模块写入devDependencies属性

### peerDependencies(“peerDependencies”:{})字段

例如：A模块是B模块的插件。用户安装的B模块是1.0版本，但是A插件只能和2.0版本的B模块一起使用。这时，用户要是将1.0版本的B的实例传给A，就会出现问题。因此，需要一种机制，在模板安装的时候提醒用户，如果A和B一起安装，那么B必须是2.0模块。

peerDependencies字段，就是用来提供指定其所需的主工具版本如：

{

“name”：“vue-webpack”,

“peerDependencies”:{

“vue”:”2.0”

}

}

(上面版本号是胡写的)

上面代码指定，当安装 vue-webpack 模块时，主程序vue必须一起安装，且版本必须是2.0，如果你的项目指定依赖的是vue的1.0版本那么就会报错

### bin字段(“bin”:{})

bin用来指定各个内部命令对应的可执行文件的位置，如：

“bin”:{

“someTool”:”./bin/someTool.js”

}

“scripts”:{

“start”:”someTool build”

}

npm run 然后按下tab建，就会显示所有可使用命令

### main字段

main指定加载的入口文件，require(‘modelName’)就会加载这个文件，这个字段的默认值是模块跟目录下的index.js文件

### config字段

config字段用于添加命令行的环境变量

### browser字段

browser指定该模板供浏览器使用的版本

### engines字段

engines 指明了该模块运行的平台，比如node的某个版本或者浏览器：

{“engines”:{“node”:”>=0.10.3<0.12”}}

### man字段

man用来指定当前模块的man文件的位置

### preferGlobal字段

preferGlobal的值是布尔值，表示当前用户不将该模块安装为全局模块时，要不要显示警告，表示该模块的本意就是安装为全局模块

### style字段

style指定供浏览器使用时，样式文件所在的位置。

### 另：版本限定：

指定版本：比如：1.2.2，表示安装指定1.2.2版本

波浪号+指定版本：比如：~1.2.2，表示安装1.2.x的最新版本（不低于1.2.2），但不安装1.3.x

插入号+指定版本：比如：^1.2.2，表示安装1.x.x的最新版本（不低于1.2.2），但不安装2.x.x

latest:安装最新版本