Package详述

1. Node.js的包通常是一些模块的集合，在模块的基础上提供了更高层的抽象，相当于提供了一些固定接口的函数库，通过定制package.json，我们可以创建更复杂、更完善、更符合规范的包用于发布。
2. Node.js在调用包时，首先会检查包中的package.json文件的main字段，将其作为包的接口模块，如果package.json文件的main字段不存在，那么Node.js会尝试寻找index.js或index.node作为包的接口。

下面是一个完整的pacage.json文件的内容，逐行解释说明：

{

"name": "testpackage", 包名。包名是唯一的，由小写字母、数字和下划线组成，不能含空格。

"description": "My package for CommonJS.", 包说明。对包进行简要描述。

"version": "0.1.0", 版本号。满足《语义化版本识别》规范的版本字符串

"keywords": [关键字数组，通常用于搜索。

"testpackage",

"liq"

],

"maintainers": [维护者数组。每个元素包含name、email(可选)、web(可选)字段。

{

"name": "liq",

"email": "liq@hotmail.com",

}

],

"contributors": [贡献者数组。格式与maintainer数组相同。包作者应该是贡献者数组的第一个元素。

{

"name": "liq",

"web": "http://blog.csdn.net/chszs"

}

],

"bugs": { 提交bug的地址，可以是网址或电邮地址。

"mail": "liq@hotmail.com",

"web": "http://blog.csdn.net/chszs"

},

"licenses": [许可证数组。每个元素要包含type(许可证名称)和url(链接到许可证文本的地址)字段。

{

"type": "Apache License v2",

"url": "http://www.apache.org/licenses/apache2.html"

}

],

"repositories": [仓库托管地址数组。每个元素要包含type(仓库类型)url(仓库地址)path(相对仓库的路径，可选)字段。

{

"type": "git",

"url": "http://github.com/chszs/packagetest.git"

}

],

"dependencies": { 包依赖。是一个关联数组，由包名和版本号组成。

"webkit": "1.2",

"ssl": {

"gnutls": ["1.0", "2.0"],

"openssl": "0.9.8"

}

}

}

除了必选字段外，规范还定义了一部分可选字段，具体如下所示。

homepage。当前包的网站地址。

os。操作系统支持列表。这些操作系统的取值包括aix、freebsd、linux、macos、solaris、vxworks、windows。如果设置了列表为空，则不对操作系统做任何假设。

cpu CPU架构的支持列表.有效的架构名称有arm、mips、ppc、sparc、x86和x86\_64。同os一样，如果列表为空，则不对CPU架构做任何假设。

engine.支持的javascript引擎列表，有效的引擎取值包括ejs、flusspferd、gpsee、jsc、spidermonkey、narwhal、node和v8。

builtin 标志当前包是否是内建在底层系统的标准组件。

directories。实现规范的列表。标志当前包实现了CommonJS的哪些规范。