lights 是FIS包管理工具。提供便捷、易用的资源安装、发布、搜索，管理工具。用于团队间快速共享资源，提升开发效率。 访问 lights.baidu.com ,找到更多资源。

**安装**

$ npm install -g lights

lights被发布为一套npm包，对它对环境的要求是：

1. 操作系统：任何能安装 nodejs 的操作系统
2. node版本：>= v0.8.0
3. 安装好lights之后，执行 lights -v，如果看到版本信息，恭喜，你已拥有lights包管理工具啦！

**特性**

* 具有CommonJS的模块化开发体验
* 管理模板、JavaScript、CSS、img等多种资源
* 完美支持FIS的模块化规范
* 大量丰富的包资源(业务组件、smarty插件，FIS demo等)
* 便捷、易用的包安装、发布、搜索
* 自动管理包依赖，无冗余依赖下载

**使用**

执行 **lights --help** 让我们来看一下lights命令的相关帮助：

Usage: lights <command>

Commands:

install install resource from lights

search search resource of lights

adduser add user of lights

publish publish resource to lights

unpublish remove resource to lights

owner change ownership of resource

config set or get config if lights

Options:

-h, --help output usage information

-v, --version output the version number

**install 下载**

$ lights install

1. 安装组件依赖。
2. 读取当前文件夹下的package.json的dependecies节点，下载所有依赖。
3. $ lights install <name>@<version>
4. Options:
5. --repos <url> : repository,仓库地址
6. lights install 下载资源以及资源的所有依赖到当前目录。
7. 不输入version时，默认下载最新版本
8. 当前目录已经存在该资源的，提示已经存在，不会覆盖，需要手动删除才能继续下载。
9. \**--repos \** ： lights默认仅与一个仓库交互，仓库之间没有同步时，可以设置--repos 参数，指定仓库下载。 例如，lights install gmu --repos fedev.baidu.com:8889
10. **repos的 url的格式为** ： **机器名或者ip地址 + 端口号。**
11. 默认的lights.baidu.com的仓库地址为 fedev.baidu.com:8889.

**search 搜索**

$ lights search <key>

1. 通过关键字搜索资源。
2. 显示相关资源的描述，仓库等。

**adduser 添加用户**

$ lights adduser

1. 通过用户名，密码，email创建用户。
2. 发布资源前需要添加用户噢

**publish 发布**

$ lights publish <folder>

1. folder: 一个包含package.json的文件夹
2. 如果包名（name）或者版本（version）已经存在，会发布失败. 可以添加 **--force** 参数来强行覆盖已存在的版本。

**package.json 配置文件**

1. 资源必须包括package.json文件
2. {
3. "name": "myproject",
4. "version": "0.0.1",
5. "description": "An example lights components",
6. "dependencies": {
7. "jquery" : "1.7.1"
8. },
9. "repository": {
10. "type": "git",
11. "url": "https://github.com/myproject"
12. },
13. "keywords": [
14. "scaffold",
15. "assets"
16. ],
17. "author": "me",
18. "license": "MIT"
19. }
20. 资源有依赖的时候，package.json添加dependencies节点。例如
21. {
22. "dependencies": {
23. "Chart" : "0.2.0",
24. "jquery" : "2.0.3"
25. }
26. }
27. 依赖的文件需要都publish到lights的仓库中，否则下载依赖会失败。依赖的组件还有依赖时，会递归下载。
28. **组件以及依赖全部下载到相同目录，没有目录嵌套**。
29. **注意：组件开发中，如果使用依赖的方式，注意相对路径的写法，保证组件下载后可用。**
30. \**目前 package.json中的版本号，仅支持具体版本号写法，不支持 >=等。否则会报错 \**

**keywords 资源分类**

前端资源聚合的资源没有强行分类，网站首页显示的资源类型，是由关键字**keywords**筛选出的。包含资源数目最多的才会显示出来噢~

默认提供以下关键字，推荐选择默认keywords，让其他同学更容易找到你的资源~ 默认keywords类型：

framework （基础库资源 —— backbone，jquery等）

css (css资源，组件等)

test (测试相关资源)

widget (组件化资源（html组件化，js组件化）

smartyPlugin (smarty插件)

assets 素材（icon，图片等）

kernal （FIS 核心资源）

monitor （监控资源 —— webspped， 统计脚本，hunter等）

scaffold （脚手架资源 —— 各种使用demo，代码框架等）

html5 （html5资源）

utils （基础util资源 —— date，array等各种操作）

**commonJS开发体验**

所谓commonJS开发体验，是基于FIS的前端框架 modJS。lights无缝结合FIS的组件化开发写法，开发的widget等组件可以在FIS中自动运行。

[具体规范细节](http://fe.baidu.com/doc/fis/2.0/user/js.text#widget)

**README.md 介绍**

1. README.md为 **markdown** 语法的文件
2. 组件资源根目录下放置README.md文件
3. 为资源添加介绍，会在网站上显示噢

**unpublish 删除**

lights unpublish <name>[@<version>]

1. 删除资源。如果不设定版本，会删除此资源的所有版本。

**owner 维护人**

lights owner ls <package name>

lights owner add <user> <package name>

lights owner rm <user> <package name>

1. **ls**: 列出对资源有修改权限的维护人。
2. **add**: 对资源添加维护人。
3. **rm**: 删除资源的维护人。

注意：所有资源的操作权限只有两种。可修改与不可修改。

**config 设置**

$ lights config set <key> <value>

$ lights config get <key>

$ lights config ls

1. **ls**: 列出lights所有的设置，包括 username，email，repos。
2. **set**: 对lights进行设置。目前仅支持设置 repos，修改username等请使用adduser命令。
3. **get**: 获取lights设置。可以根据key值来获取。

**update 更新**

$ lights update <pkg>

1. 更新资源到最新版本。
2. **update会覆盖更新**。原有已经下载的包会被覆盖，包括所有依赖。

**remove 删除**

$ lights remove <pkg>

1. 删除已经安装的资源。删除目录为命令行所在的目录。
2. **remove不会删除依赖资源**。删除依赖可能会导致其他依赖的资源不可用，remove仅支持删除资源本身

**设置仓库**

lights支持分布式的仓库存储。自行搭建的lights私有仓库，需要在lights中设置repos的url来指定。

$lights config set repos fedev.baidu.com:8889

1. 默认的lights.baidu.com的仓库地址为 fedev.baidu.com:8889.
2. repos **url的格式为** : **机器名或者ip地址 + 端口号**。
3. 分布式的仓库支持，后续会在仓库页面，设置数据同步的功能。