介绍fis-plus 开发命令，完成你的所有开发需求

执行 **fisp --help** 让我们来看一下fis命令的相关帮助：

Usage: fis-plus <command>

Commands:

release build and deploy your project

install install components and demos, create `widget` and `module`

server launch a php-cgi server

Options:

-h, --help output usage information

-v, --version output the version number

--no-color disable colored output

* **fisp release**： 编译并发布你的项目
* **fisp server**：启动一个 **1.8M** 大小的内置调试服务器，它采用php-java-bridge技术实现， 依赖java、php-cgi外部环境，可以 **完美支持运行php程序** 哦。

## fis release

release是一个非常强大的命令。

* 它的主要任务就是进行代码的 **编译** 与 **部署**
* 它的参数囊括了前端开发所需的各种基础功能。通过添加不同参数，组合使用。

可以在命令行下执行 执行 **fisp release -h** 让我们来看一下fis命令的相关帮助：

Usage: release **[**options**]**

Options:

-h, --help output usage information

-d, --dest <names> release output destination

-m, --md5 **[**level**]** md5 release option

-D, --domains add domain name

-l, --lint with lint

-t, --test with unit testing

-o, --optimize with optimizing

-p, --pack with package

-w, --watch monitor the changes of project

-L, --live automatically reload your browser

-c, --clean clean compile cache

-r, --root <path> set project root

-f, --file <filename> set fis-conf file

-u, --unique use unique compile caching

--verbose enable verbose output

* --domain 或 -D 参数，对静态资源添加domian。[domain配置](http://oak.baidu.com/docs/fis-plus/user/config.html#domain%E9%85%8D%E7%BD%AE)
* --watch 或 -w 参数，对项目进行增量编译，监听文件变化再触发编译
* --md5 [level] 或 -m [level] 参数，在编译的时候可以对文件自动加md5戳，从此告别在静态资源url后面写?version=xxx的时代
* --lint 或 -l 参数，支持在编译的时候根据项目配置自动代码检查
* --test 或 -t 参数，支持在编译的时候对代码进行自动化测试
* --pack 或 -p 参数，对产出文件根据项目配置进行打包
* --optimize 或 -o 参数，对js、css、html进行压缩
* --dest [path|name] 或 -d 参数，来指定编译后的代码部署路径，支持发布到 **本地目录、本地调试服务器目录、远程机器目录(需要配置)**，它与--watch参数配合使用，可以让你的代码保存就上传！而且--dest值支持逗号分隔，这也就意味着，你 **一次编译可以同时发布到本地以及多台远程机器上**！举几个栗子:
  + 发布到fisp server open目录下用于本地调试
  + fisp release
  + *# or*
  + fisp release --dest preview
  + 发布到项目根目录的output目录下， 注意，这里的output其实是一个内置的部署配置名，而不是一个目录名。
  + fisp release -d output
  + 发布到相对 工作目录 的路径
  + fisp release -d ../output
  + 发布到绝对路径
  + fisp release -d /home/work/ouput
  + *# win*
  + fisp release -d d:/work/output
  + 使用配置文件的 [deploy节点配置](http://oak.baidu.com/docs/fis-plus/user/deploy.html) 进行发布，此配置可将代码上传至远端
  + fisp release -d remote
  + 以上所有发布规则任意组合使用（一次编译同时上传到多台远端机器 & 项目根目录下的output & 调试服务器根目录 & 本地绝对路径）
  + fisp releaes -d remote,qa,rd,output,preview,D:/work/output
* --live 或 -L 参数，支持编译后自动刷新浏览器。fis需要使用 [LiveReload浏览器扩展](http://feedback.livereload.com/knowledgebase/articles/86242-how-do-i-install-and-use-the-browser-extensions-) 来连接fis的livereload服务器：
  + [Safari](http://download.livereload.com/2.0.9/LiveReload-2.0.9.safariextz)
  + [Chrome](https://chrome.google.com/webstore/detail/livereload/jnihajbhpnppcggbcgedagnkighmdlei)
  + [Firefox](http://download.livereload.com/2.0.8/LiveReload-2.0.8.xpi)
  + [IE](https://github.com/dvdotsenko/livereload_ie_extension/downloads) - 真应该滚粗地球

安装好浏览器插件后，在common模块基础模板中添加一段注释代码： <!--添加livereload注释--> <!--livereload-->然后在相应需要开启livereload的模块(只能同时开启一个)编译时添加-L或--live参数，在浏览器里点击插件按钮进行连接，就完成了设置可以享受了~

* --unique 或 -u 参数，每次编译创建一个独立的缓存目录，**解决同一台机器多人编译互相影响的问题**。如果在自己机器一人编译，那就不需要了。
* --verbose 参数，编译时输出log信息
* --lint 或 -l 参数，对项目进行lint检查
* --test 或 -t 参数，对项目执行自动化测试。目前仅保留接口，没有测试功能
* --clean 或 -w 参数，对项目进行清缓存编译。
* --root 或 -r 参数，设置项目根路径
* --file 或 -f 参数，编译时指定fis-conf配置文件路径

## fis server

提供本地调试的轻量级服务器，完全模拟线上运行，用户可通过配置测试数据、请求模拟等，进行线下无后端环境开发。

fis的调试服务器依赖于用户本地的 **jre** 和 **php-cgi** 环境

* [环境安装请移步这里](http://oak.baidu.com/docs/fis-plus/user/quickstart.html)

搞定环境后，让我们来启动调试服务器看看：

$ fisp server start

checking java support : version 1.6.0

checking php-cgi support : version 5.2.11

starting fis-server on port : 8080

服务器启动之后，它会自动检查环境，最后告诉你它监听了8080端口，这个时候，你的浏览器应该打开了一个调试服务器根目录的浏览页面，地址是 [**http://localhost:8080/**](http://localhost:8080/)。

常用server命令:

//启动服务默认端口8080

fisp server start

//停止服务

fisp server stop

//启动服务设置端口为8081,同时设置php-cgi路径

fisp server start -p 8081 --php\_exec "D:/php/php-cgi.exe"

//打开本地预览环境

fisp server open

//清空本地预览环境,<慎用!>，清空后请安装本地调试环境

fisp server clean

更多使用，在命令行执行 fisp server -h 查看

## fisp install

lights是FIS提供包管理工具，托管所有FIS资源。fisp install集成lights，并提供脚手架功能。

Usage: install <command> **[**options**]**

Commands:

module create a module

widget create a widget

Options:

-h, --help output usage information

-s, --scaffold <scaffold>

-d, --dir <name> create to dir

--with-plugin **if** create a module, whether include `plugin`

--repos <url> repository

--verbose output verbose help

--list **[**query**]** list component from the repos

--ld <left\_delimiter> smarty left\_delimiter

--rd <right\_delimiter> smarty right\_delimiter

Examples:

$ fisp install module -d ./to/directory/other

$ fisp install module -d ./to/directory/common --with-plugin

$ fisp install module -d ./to/directory/other --ld '<%' --rd '%>'

$ fisp install widget -d ./widget/box

$ fisp install modjs //just download 'modjs'