**命令行工具**

**为什么要使用命令行？**

**更高效**

无反馈的系统比有反馈效率更高，

**可以和其他命令协同工作**

这符合unix的哲学：每个命令都很简单然后让他们工作在一起，可以使用管道，或者cron定时任务，自动化我们的工作。或者将你的editplus,vim,emacs,zend等等编辑器同他结合在一起。

**更丰富的应用场景**

批量更新上百台服务器集群里的配置？或者自动化，或者经过N级ssh隧道操作远程系统？命令行才是最具有适应力的交互系统。只有字符才能完全控制计算机。

**怎样使用命令行(linux下面在cmd前面加上php window不用)**

应用base下面的cmd; 如果是windows,那就是cmd.bat;如果是linux，在命令前需要加php，比如：php app/base/cmd help -v。

输入app/base/cmd会进入交互shell模式，或者直接输入某个命令：app/base/cmd xxxx。

你可以使用app/base/cmd help -v看到所有支持的命令列表，-v会显示各个命令所支持的参数。

app/base/cmd help

exit 退出

man 显示帮助

sh 执行操作系统命令

mkconfig 创建config文件

ls 列出所有应用

cd 切换当前应用

install 安装应用

uninstall 卸载应用

update 升级应用程序

trace 打开/关闭性能检测

status 显示系统状态

search 在程序库中搜索

createproject 创建新项目

kvrecovery kvstore数据恢复

cacheclean 清除缓存

configcompat config兼容配置检测

应用提供的命令： -------------------------------------------------------

base:queue list 列出所有队列任务

base:queue flush 立即执行所有队列任务

base:queue exec 执行指定的队列任务

base:queue clear //todo 清除所有队列任务

base:queue active //todo 激活某任务

base:queue disable //todo 暂停某任务

base:task list 列出所有计划任务

base:task exec 按计划执行任务

dev:check check //todo 执行代码检查

dev:doc update 执行测试用例

dev:new app 创建一个app

dev:project create 创建新项目

dev:show services 显示注册的Service

dev:show viewtags 显示所有可用模板标签

dev:show classfile 显示类的文件地址

dev:show depends 生成已安装的app依赖关系图, Graphviz格式

dev:test do 执行测试用例

原生php命令 ----------------------------------------------------------------

输入命令如果以分号[;]结尾，则被认为是一条php语句. 例如:

1> $a = 2;

int(2)

2> pow($a,8);

int(256)

**几个例子**

**谁提供的模版插件**

app/base/cmd dev:show services | grep "^view\_helper"

view\_helper base\_view\_helper

view\_helper ectools\_view\_helper

view\_helper desktop\_view\_helper

view\_helper b2c\_view\_helper

view\_helper site\_view\_helper

**看看app的依赖关系**

app/base/cmd dev:show depends | dot -Tjpg -odepends.jpg

嗯， 我这里真的装了很多应用 :)

sh: dot: command not found

**交互模式**

如果你仅仅输入app/base/cmd, 就会进入交互模式：

Ecos shell V1.3 (abort with ^C), Root: /Users/wanglei/Sites/ecstore

1> ls

b2c active

base active

content active

dbeav active

desktop active

dev active

ectools active

extdb active

image active

jiankongbao active

pam active

site active

varnish active

2> trace on

Trace mode is on

3> cacheclean

Cache Clear...

...Clear Success,ok

\* Command memroy useage = 512, Time left = 0.305762

4> trace off

Trace mode is off

5> exit

exit

**编写一个新命令**

**简单例子**

编写一个新命令非常简单，在你的app里创建lib/command目录，然后新建一个文件app/myapp/lib/command/hello.php。

 <?php  
class myapp\_command\_hello extends base\_shell\_prototype{  
  
    var $command\_world = 'Helloworld';  
    function command\_world(){  
        echo "hello, world\n";  
    }  
  
}

app/base/cmd myapp:hello world

hello, world

当存在同名的command\_开头的函数和属性时，函数将作为该命令执行的实体，属性会作为该命令的描述。

**命令需要参数**

使用者空格分割的输入项，将作为该函数的调用参数。

 <?php  
class myapp\_command\_hello extends base\_shell\_prototype{  
  
    var $command\_3guys = 'Hello 3guys';  
    function command\_3guys($one, $two, $three){  
        echo "1\t", $one,"\n";  
        echo "2\t", $two,"\n";  
        echo "3\t", $three,"\n";  
    }  
  
}

输出：

app/base/cmd myapp:hello 3guys jack mike tom

1 jack

2 mike

3 tom

更多时候，我们无法判断使用者真实提交了几个参数，所以请给参数制定默认值，或者使用[func\_get\_args()](http://php.net/manual/en/function.func-get-args.php)获取各个参数。

**命令可以使用选项**

 <?php  
class myapp\_command\_hello extends base\_shell\_prototype{  
  
    var $command\_world = 'Helloworld';  
    var $command\_world\_options = array(  
            'quietly'=>array('title'=>'项目模板','short'=>'q'),  
            'from'=>array('title'=>'项目模板','need\_value'=>1),  
        );  
    function command\_world(){  
        $options = $this->get\_options();  
        if($options['quietly']){  
            echo "hello, xxx\n";  
        }else{  
            echo "hello, world\n\t\t --from {$options['from']}\n";  
        }  
    }  
  
}

app/base/cmd myapp:hello world --from tom

hello, world

--from tom

这里使用-q或者--quietly是等价的。

**输出修饰**

1. $this->output\_table($rows) 即可将数组输出格式化好的表格。
2. $this->output\_line($string) 可以输出一个分行。

 <?php  
class myapp\_command\_hello extends base\_shell\_prototype{  
  
    var $command\_cart = 'Hello guys';  
    function command\_cart(){  
        $this->output\_line("Cart");  
  
        $rows = array();  
        $rows[] = array('Type','Price',"Number");  
        $rows[] = array('apple','1.3',20);  
        $rows[] = array('orange','3',20);  
        $rows[] = array('banana','4.5',10);  
        $rows[] = array('something very long, very long,long','4.5',10);  
        $this->output\_table($rows);  
    }  
  
}

效果：

app/base/cmd dev:hello cart

Cart --------------------------------------------------------

Type Price Number

apple 1.3 20

orange 3 20

banana 4.5 10

something very long, very long,long 4.5 10