

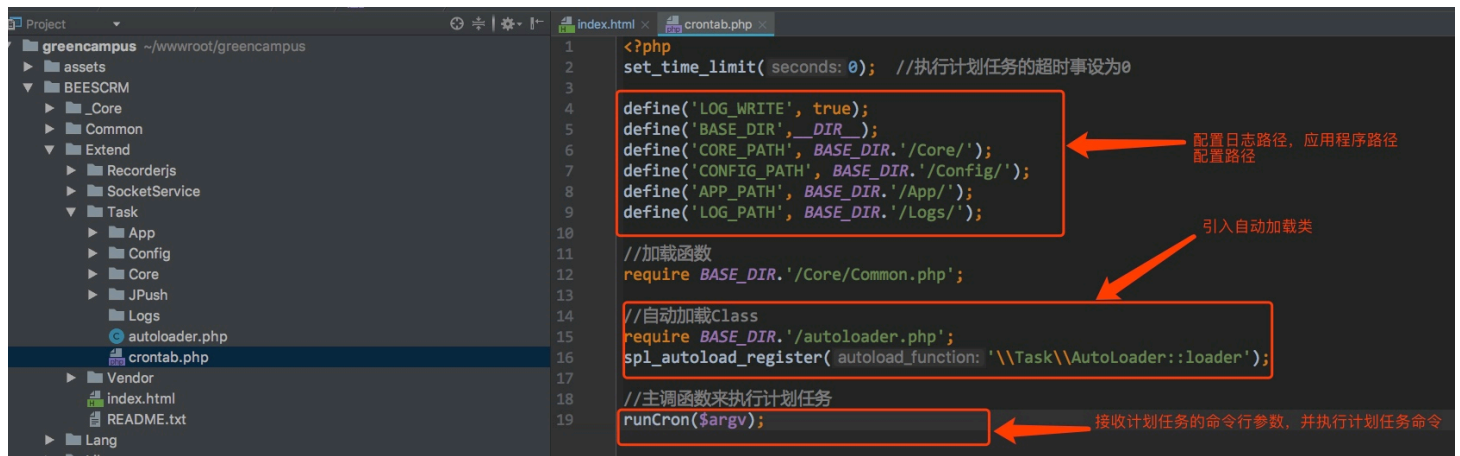
# crontab框架

- crontab框架
  - 目录结构说明
  - 入口文件说明
  - 计划任务命令行
  - 计划任务代码编写
    - 1. 创建模块（非必要）
    - 2. 创建业务类（必要）
    - 3. 创建方法（必要）
    - 4. 参数（非必要）
  - 数据库操作
    - 1. 数据库如何增删查改
    - 2. 数据库如何连接
  - 注意说明
    - 1.
    - 2.
    - 3.

## 目录结构说明

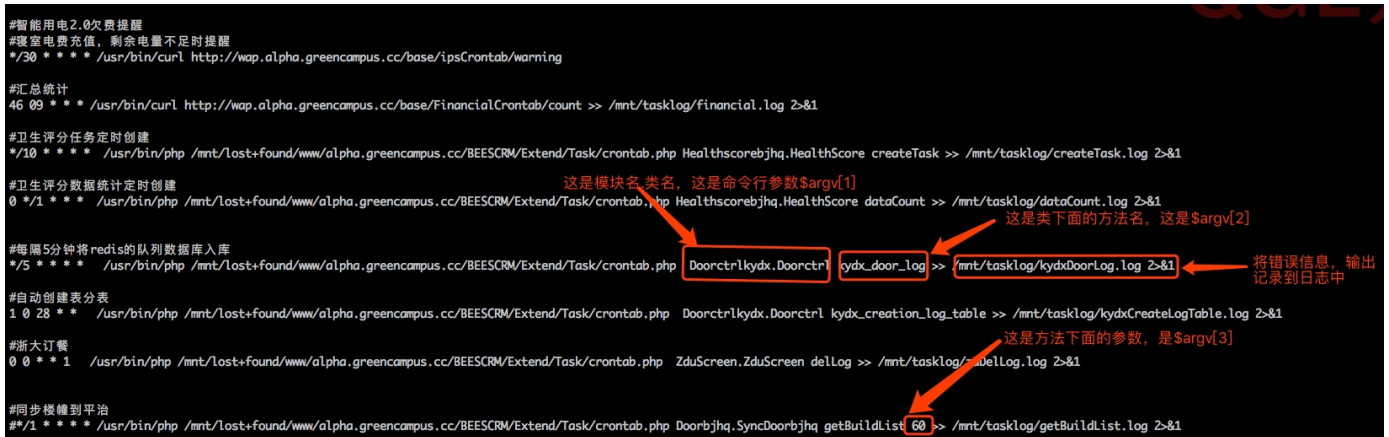


## 入口文件说明



# 计划任务命令行

1. 命令行参数说明
2. 框架业务层在命令行中的使用



# 计划任务代码编写

根据入口文件中的runCron函数，来对命令行接收的参数进行处理可得知业务模块、业务类，方法，参数之间的关系

```
function runCron($commandArgv)
```

```
{  
    if (!isset($commandArgv[1])) {  
        exit("class's param is missing");  
    }  
  
    if (!isset($commandArgv[2])) {  
        exit("method's param is missing");  
    }  
}
```

类名+方法名不能为空的

```
//判断类是否存在
```

```
$classNameStr = $commandArgv[1];  
if (false === strpos($classNameStr, needlet: '.')) {  
    $class = '\\Task\\App\\' . $classNameStr;  
} else {  
    $classNameArr = explode( delimiter: '.', $classNameStr);  
    $class = '\\Task\\App\\' . $classNameArr[0] . '\\'. $classNameArr[1];  
}  
$class = ucfirst($class);  
if (!class_exists($class)) {  
    exit("class {$class} is not exists \n");  
}
```

业务处理复杂的时候需要用到模块，简单的业务可以直接用类来处理，模块和类用.进行分割

```
//判断类中的方法是否存在
```

```
$execObj = new $class();  
$func = $commandArgv[2];  
if (!method_exists($execObj, $func)) {  
    exit("method {$func} is not exists \n");  
}
```

判断方法，方法是必须要存在

```
//设置方法的参数
```

```
$params = array();  
foreach ($commandArgv as $key => $param) {
```

```
//判断类中的方法是否存在
```

```
$execObj = new $class();  
$func = $commandArgv[2];  
if (!method_exists($execObj, $func)) {  
    exit("method {$func} is not exists \n");  
}
```

```
//设置方法的参数
```

```
$params = array();  
foreach ($commandArgv as $key => $param) {  
    if ($key > 2) { //将参数设置到数组中去  
        $params[] = $param;  
    }  
}
```

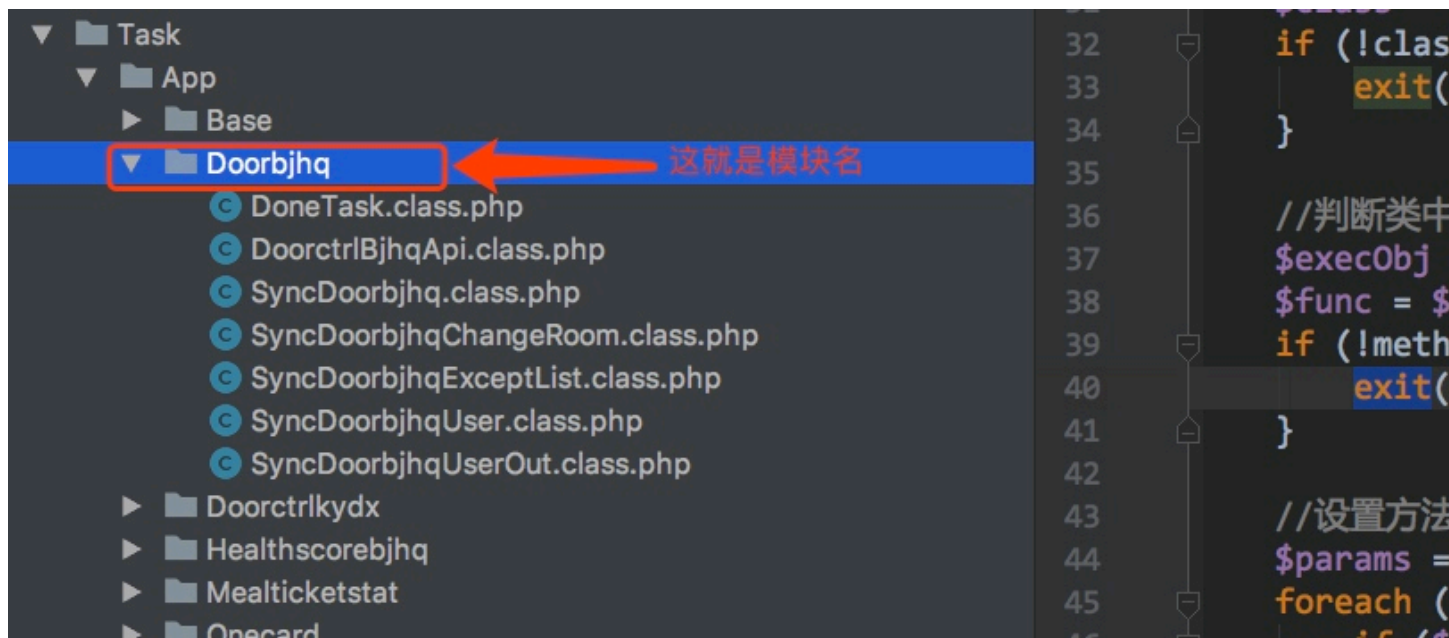
这是方法中的参数列表，可以为空

```
//执行类中的方法
```

```
call_user_func_array(array($execObj, $func), $params);
```

去执行类下面的方法

## 1. 创建模块（非必要）



## 2. 创建业务类（必要）

业务类往往都要去继承Base基础类,Base基础类里面来实现数据库连接，日志类的声明等公共的操作。



## 3. 创建方法（必要）

类似于Controller层的使用



//拉取redis中存储的同步入住任务的数据

public function pullTaskData()

```
{
    $redis = RedisQueueService::getInstance();
    $returnData = $redis->pull( listKey: 'bjhq_crontab');
    if(!$returnData) {
        exit("已经没有任务了! \n");
    }

    $returnData = json_decode($returnData, assoc: true);
    $className = $returnData['className'];
    $methodName = $returnData['methodName'];
    $params = $returnData['params'];
    if($params){
        $params = json_decode($params, assoc: true);
        if(is_array($params)){
            $paramStr = implode( glue: ' ', $params);
        }else{
            $paramStr = $params;
        }

        $execStr = "/usr/bin/php /mnt/lost+found/www/alpha.greencampus.cc/BEESCRM/Ext
    }else{
        $execStr = "/usr/bin/php /mnt/lost+found/www/alpha.greencampus.cc/BEESCRM/Ext
    }
}
```

## 4. 参数（非必要）

如果命令行需要传参数，就需要在方法中传参数

```
/**
 * 手动同步到平治楼幢信息
 * @param $account_id
 */
public function getBuildList($account_id){
    $sql = "select b.building_id,b.community_id,b.building_name,a.community_name from gc_buildin
    $result = $this->mysql->query($sql)->fetchAll(PDO::FETCH_ASSOC);

    $doorctrlBjqApi = new DoorctrlBjqApi();

    $functionName = 'AddUpdateDepart';
    foreach($result as $k=>$v) {
        $params = array(
            'depart' => array('departName' => $v['community_name'].$v['building_name'], 'outID'
        );

        try{
            $aa = $doorctrlBjqApi :: callApi($functionName, $params);
        }catch (\Exception $e) {
            $msg = $e->getMessage();
            error_log("ms:{json_encode($msg)}",3,"/mnt/tasklog/pullTaskData.log");
        }

        //
        //
        //
        cdump(json_encode($aa).$v['building_name']);
        cdump(json_encode($params));exit;
    }
}
```

# 数据库操作

---

## 1. 数据库如何增删查改

数据库操作是采用PDO操作，PDO类是使用medoo 查询一律用原生SQL编写，数据批量入库，可以采用medoo自带的方法insert("表名", "二维数组")。数据做update操作时，也是可以用其update操作

## 2. 数据库如何连接

数据库支持多个并且是多种数据库的连接方式，像mysql,pgsql,oracle等。并且只要在Config目录的db.php下配置多个数据库连接即可，在使用的时候，可以用Base类下selectDb("db.php下配置数组的key")

## 注意说明

---

### 1.

本框架是独立于web项目的框架，不依赖于web项目，可以单独部署。

### 2.

命令行执行的时候，不能接收POST,GET,SESSION,COOKIE,SERVER等web端的全局变量。它接收参数都是通过\$argv全局变量。

### 3.

命令行执行，不能像web端一样用相对路径(../,./), 只能用绝对路径。