thinkphp+queue+supervisor

作者：xiaoxiezaijia
原文链接：https://ld246.com/article/1569398569804
来源网站：链滴

1.thinkhphp安装queue扩展

我使用的是composer安装（composer安装地址：https://pkg.phpcomposer.com）

直接在项目根目录运行 composer require topthink/think-queue

2.配置queue

安装完后会在config文件下生成一个queue.php文件，配置项如下：（使用的是redis驱动，服务器是linux,安装很方便，yum install redis）

```
return [
    'connector'  => 'Redis',        // Redis 驱动
    'expire'     => 60,            // 任务的过期时间，默认为60秒; 若要禁用，则设置为 null
    'default'    => 'default',      // 默认的队列名称
    'host'       => '127.0.0.1',    // redis 主机ip
    'port'       => 6379,           // redis 端口
    'password'   => '',             // redis 密码
    'select'     => 0,             // 使用哪一个 db，默认为 db0
    'timeout'    => 0,              // redis连接的超时时间
    'persistent' => false,          // 是否是长连接
];
```

3.创建任务

单模块项目推荐使用 app\job 作为任务类的命名空间 多模块项目可用使用 app\module\job 作为任类的命名空间 也可以放在任意可以自动加载到的地方

任务类不需继承任何类，如果这个类只有一个任务，那么就只需要提供一个fire方法就可以了，如果多个小任务，就写多个方法，下面发布任务的时候会有区别

每个方法会传入两个参数 think\queue\Job $job （当前的任务对象） 和 $data（发布任务时自定义数据）

还有个可选的任务失败执行的方法 failed 传入的参数为$data（发布任务时自定义的数据）

```
namespace app\job;

use think\queue\Job;

class Job1{

   public function fire(Job $job, $data){

          //....这里执行具体的任务

           if ($job->attempts() > 3) {
               //通过这个方法可以检查这个任务已经重试了几次了
           }
          //如果任务执行成功后 记得删除任务，不然这个任务会重复执行，直到达到最大重试次数后败后，执行failed方法
           $job->delete();
```

原文链接：thinkphp+queue+supervisor

```
            // 也可以重新发布这个任务
            $job->release($delay); //$delay为延迟时间

    }
    public function failed($data){

        // ...任务达到最大重试次数后，失败了
    }

}
```

4.发布任务

think\Queue::push($job, $data = '', $queue = null) 和 think\Queue::later($delay, $job, $data = '', $queue = null) 两个方法，前者是立即执行，后者是在 $delay 秒后执行

$job是任务名

单模块的，且命名空间是 app\job 的，比如上面的例子一,写Job1类名即可

多模块的，且命名空间是 app\module\job 的，写 model/Job1 即可

如果一个任务类里有多个小任务的话，需要用 @+方法名app\lib\job\Job2@task1、app\lib\job\Jo2@task2

$data 是你要传到任务里的参数

$queue 队列名，指定这个任务是在哪个队列上执行，同下面监控队列的时候指定的队列名,可不填

5.安装supervisor

yum install supervisor

6.配置

安装好后在/etc/会生成一个supervisord.conf文件及一个supervisord.d文件目录

supervisord.conf是一些默认配置，可自行修改：

```
[unix_http_server]
file=/tmp/supervisor.sock   ;UNIX socket 文件，supervisorctl 会使用
;chmod=0700                 ;socket文件的mode，默认是0700
;chown=nobody:nogroup       ;socket文件的owner，格式：uid:gid

;[inet_http_server]         ;HTTP服务器，提供web管理界面
;port=127.0.0.1:9001        ;Web管理后台运行的IP和端口，如果开放到公网，需要注意安全性
;username=user              ;登录管理后台的用户名
;password=123               ;登录管理后台的密码

[supervisord]
logfile=/tmp/supervisord.log ;日志文件，默认是 $CWD/supervisord.log
logfile_maxbytes=50MB        ;日志文件大小，超出会rotate，默认 50MB，如果设成0，表示不限大小
logfile_backups=10           ;日志文件保留备份数量默认10，设为0表示不备份
```

原文链接：thinkphp+queue+supervisor

```
loglevel=info                ;日志级别，默认info，其它: debug,warn,trace
pidfile=/tmp/supervisord.pid ;pid 文件
nodaemon=false               ;是否在前台启动，默认是false，即以 daemon 的方式启动
minfds=1024                  ;可以打开的文件描述符的最小值，默认 1024
minprocs=200                 ;可以打开的进程数的最小值，默认 200

[supervisorctl]
serverurl=unix:///tmp/supervisor.sock ;通过UNIX socket连接supervisord，路径与unix_http_ser
er部分的file一致
;serverurl=http://127.0.0.1:9001 ; 通过HTTP的方式连接supervisord

; [program:xx]是被管理的进程配置参数，xx是进程的名称
[program:xx]
command=/opt/apache-tomcat-8.0.35/bin/catalina.sh run  ; 程序启动命令
autostart=true       ; 在supervisord启动的时候也自动启动
startsecs=10         ; 启动10秒后没有异常退出，就表示进程正常启动了，默认为1秒
autorestart=true     ; 程序退出后自动重启,可选值：[unexpected,true,false]，默认为unexpected
表示进程意外杀死后才重启
startretries=3       ; 启动失败自动重试次数，默认是3
user=tomcat          ; 用哪个用户启动进程，默认是root
priority=999         ; 进程启动优先级，默认999，值小的优先启动
redirect_stderr=true ; 把stderr重定向到stdout，默认false
stdout_logfile_maxbytes=20MB  ; stdout 日志文件大小，默认50MB
stdout_logfile_backups = 20   ; stdout 日志文件备份数，默认是10
; stdout 日志文件，需要注意当指定目录不存在时无法正常启动，所以需要手动创建目录（superviso
d 会自动创建日志文件）
stdout_logfile=/opt/apache-tomcat-8.0.35/logs/catalina.out
stopasgroup=false     ;默认为false,进程被杀死时，是否向这个进程组发送stop信号，包括子进程
killasgroup=false     ;默认为false，向进程组发送kill信号，包括子进程

;包含其它配置文件
[include]
files = relative/directory/*.ini    ;可以指定一个或多个以.ini结束的配置文件
```

最后一行，个人习惯命名conf，改为

```
[include]
files = relative/directory/*.conf
```

/etc/supervisord.d目录用来存放用户自定义的进程配置，参考：

```
command=php think queue:work --queue RecordJob --daemon ;执行命令
directory=/home/phpweb/gwjkj/ ;执行命令目录
user=phpweb ;执行命令用户
stdout_logfile=/opt/supervisor_log/run.log ;日志位置
autostart=true ;
autorestart=true ;
startsecs=60 ;
stopasgroup=true ;
ikillasgroup=true ;
startretries=1 ;
redirect_stderr=true ;
```

7.启动服务

service supervisord start 启动服务
service supervisord stop 终止服务
service supervisord restart 重启服务

原文链接：thinkphp+queue+supervisor