在全局服务下注册一个 自定义的进程

use EasySwoole\EasySwoole\ServerManager;

use App\Lib\Process\ConsumerTest;

use App\Lib\Redis\Redis;

use EasySwoole\Component\Di;

...............

public static function mainServerCreate(EventRegister $register)

{

//注册redis进容器

Di::getInstance()->set('REDIS',Redis::getInstance());

$allNum = 3;

for ($i = 0 ;$i < $allNum;$i++){

//开启了是哪个子进程,自定义 进程名 , 去调用 ConsumerTest 里面的方法

ServerManager::getInstance()->getSwooleServer()->addProcess((new ConsumerTest("my\_dingyi\_consumer\_{$i}"))->getProcess());

}

}

要执行的文件(new ConsumerTest ) 这个文件

<?php

namespace App\Lib\Process;

use EasySwoole\Component\Process\AbstractProcess;

use Swoole\Process;

use EasySwoole\Component\Di;

class ConsumerTest extends AbstractProcess

{

private $isRun = false;

public function run($arg)

{

// TODO: Implement run() method.

/\*

\* 举例，消费redis中的队列数据

\* 定时500ms检测有没有任务，有的话就while死循环执行

\*/

$this->addTick(500,function (){

if(!$this->isRun){

$this->isRun = true;

//$redis = new \redis();//此处为伪代码，请自己建立连接或者维护redis连接

while (true){

try{

//容器取出

$task = Di::getInstance()->get('REDIS')->lPop('name\_list');

if($task){

// 发短信,发邮件,发消息的逻辑,写日志log

}else{

break;

}

}catch (\Throwable $throwable){

break;

}

}

$this->isRun = false;

}

var\_dump($this->getProcessName().' task run check');

});

}

public function onShutDown()

{

// TODO: Implement onShutDown() method.

}

public function onReceive(string $str, ...$args)

{

// TODO: Implement onReceive() method.

}

}