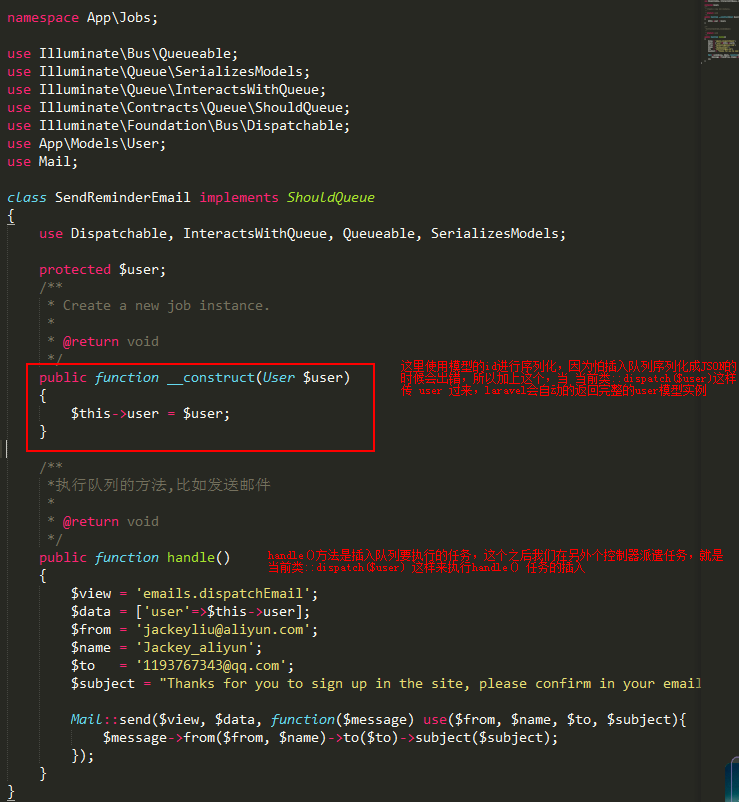
1. Laravel自带的queue队列：(在配置Email的时候，记得要添加来自哪里)
2. 配置文件：config/queue.php：



1. 现在可以创建队列啦： **php artisan make:job** SendEmail(这个名字自定义)，默认是会在app/Jobs/目录下生成：



1. 队列的准备好啦，现在我们就要来执行任务的插入调用啦：



1. 我们在网页访问3）这个控制器的方法后，它就会插入到队列，但不会执行，我们需要执行队列任务的命令: **php artisan queue:**

queue:work redis默认只执行一次队列请求, 当请求执行完成后就终止；

queue:work redis --daemon同listen一样，不同的是work不需要再次加载框架，直接运行任务，一般推荐使用这个来处理队列监听。

注：使用 queue:work redis --daemon ，当更新代码的时候，需要停止，然后重新启动，这样才能把修改的代码应用上。

额外功能：

1. 延迟任务： dispatch($user)->delay(Carbon::now()->addMinutes(10));//延迟10分钟
2. 超时关闭任务(但先执行最大尝试数) 在我们创建的handle()类中 public $timeout = 120;
3. 在我们创建的handle()类中 public $tries = 5;//队列执行失败，尝试执行5次;
4. Php artisan queue:work --sleep=3 意思是如果队列有数据则没影响，继续执行，如果没有数据，则队列睡眠3秒
5. 还有timeout的时候一定要比retry\_after短几秒，这样就能保证任务进行总能在失败重试前就被杀死，如果大于的话，则任务进程会被执行两次。

6.如果执行失败则删除的（也就是执行一次），在handle()方法里的第一行写入 $this->job->delete();

.env

