

查看messenger.yaml配置文件, 在transports配置下有个failed传输, failed传输使用了doctrine传输, 队列的名称是failed。我们取消行前的注释, 在第4行messenger组件提供了failure\_transport配置。我们取消第4行行前的注释, 当consume命令行消费一条消息时, 可能会出现一些错误。当出现错误时, 命令行会向传输中发送一条失败的消息, 传输队列即将失败的消息调度到指定的失败传输队列中, 现在我们配置了两个transports。

我们打开控制台, 重新启动一下consume命令行, 输入-vv, 现在提示让我们选择一个传输队列。我们选择0。回到浏览器, 我们来退出mailcatcher, 让邮件发送失败。点击退出。

回到管理端, 我们新增一篇文章, 文章的标题叫做Hello Failed, 点击添加回到控制台。现在命令行就收到了消息, 等待10秒钟之后, 提示了一条错误。然后再等待1秒钟之后进行第1次重试。现在再次获取消息, 在等待2秒钟之后进行了第2次重试。等待4秒钟之后进行了第3次重试, 然后进行第4次获取消息, 现在提示我们, 当前的消息已经被拒绝了。并且会将消息发送到failure transport。

我们打开数据库客户端, 查看一下数据库中的消息, 当前消息就是我们发送失败的消息。但是队列的名称并不是之前的名称了, 现在队列的名称修改为了failed。

回到项目, 我们可以查看一下messenger提供的所有命令行。输入symfony console list messenger, messenger组件提供的命令行中有三条是关于发送失败时的命令行, 我们可以使用messenger:failed:show来查看一下队列中所有失败的消息。

输入symfony console, 粘贴, 当前队列中有一条失败的消息。我们可以使用messenger:failed:retry, 重新尝试发送这条消息。输入symfony console, retry。

我们当前没有启动mailcatcher, 它提示我们需要重新尝试吗, 我们输入yes, 现在仍然会发送失败, 直接就退出了命令行。

我们启动mailcatcher, 重新执行命令行, 输入yes。这次没有报错。回到浏览器, 查看邮件列表, 我们收到了新的邮件。回到项目, 我们查看一下失败时处理的流程。当消息处理失败时, 它会进行几次重试, 每次重试时都会进行一个时间的延迟, 我们可以查看一下messenger配置文件, 所有可以配置的配置选项。

打开新的控制台输入symfony console config:dump framework messenger, 我们查看transports配置, 首先我们可以设置transports的名称, 然后设置dsn。

在下面有个重试策略, 最大的重试次数, 默认值为3。每次重试时都会有一个延迟, 这里设置为1秒, 如果重试次数大于1, 那么在每次进行重试时, 都会在延迟时间上乘以2。我们可以修改配置来调整重试的次数以及延迟的时间。

我们来修改async配置, 注释这行代码, 添加async, 设置dsn。dsn还使用之前的配置。然后我们设置重试策略, 复制配置项, 粘贴。最大的重试次数, 我们修改一下, 这里修改为5。延迟时间, 这里仍然设置为1秒钟, 延迟时间的乘数, 这里我们设置为3。

```
#config/packages/messenger.yaml

framework:
    messenger:
        # Uncomment this (and the failed transport below) to send failed
        messages to this transport for later handling.
        failure_transport: failed
```

```
    transports:
        # https://symfony.com/doc/current/messenger.html#transport-configuration
        #
        async: '%env(MESSENGER_TRANSPORT_DSN)%'
        async:
            dsn: '%env(MESSENGER_TRANSPORT_DSN)%'
            retry_strategy:
                max_retries: 5
                delay: 1000
                multiplier: 3
            failed: 'doctrine://default?queue_name=failed'
        # sync: 'sync://'

    routing:
        # Route your messages to the transports
        'App\Message\SendMessage': async
```

我们修改了messenger.yaml配置文件，每次修改messenger.yaml配置文件之后，我们都要暂停之前的consume命令行，然后再重新启动命令行。messenger提供了一个命令行来暂停所有的工作者进程，我们复制**messenger:stop-workers**命令行。

输入**symfony console**，粘贴，现在就暂停了所有的工作者进程，我们重新启动consume命令行。选择async传输队列，我们退出mailcatcher，点击确定。新增一篇文章，点击添加，我们查看底部控制台。

在等待10秒钟之后进行了第1次重试，然后进行第2次重试，重试的间隔就变成了3秒。现在consume命令行会按照我们的配置进行自动重试，consume命令行可以叫做工作者进程，在进行5次尝试之后会将失败的消息发送到失败队列。

我们的配置已经生效了，在下节课，我们来学习常用的AMQP队列。