我们搜索symfony messenger，查看symfony messenger文档。我们先来查看一下messenger组件的介绍，messenger组件提供了一个消息总线，能够发送消息，然后在应用程序中立即处理它们(消息)，或者通过传输比如队列发送它们(消息)，以便稍后处理。

如果使用消息队列发送异步消息，我们就需要使用到messenger组件，首先这里提示我们需要安装组件，安装完成组件之后我们需要定义一个消息类，这个消息类中就是我们需要发送的消息内容。然后我们需要定一个消息处理类，消息处理类需要实现MessageHandlerInterface。在Handler类中我们需要实现__invoke方法。

然后在controller方法中，我们使用消息总线$bus对象来调度我们的消息。将消息发送到对应的传输或者队列上，然后消息处理器Handler来获取传输或者队列上的消息进行处理，用这个流程来实现消息的异步处理。

往上看，我们先安装messenger组件，回到项目。打开控制台，安装完成后命令行有一些提示，首先让我们取消.env文件中的注释。我们查看.env文件，在.env文件中添加了一些消息队列的transport。我们可以根据需要使用doctrine队列，或者AMQP队列，或者redis队列。

然后我们查看一下config目录，在config目录中增加了messenger.yaml配置文件，我们可以在messenger配置文件中，为不同的消息指定不同的传输队列，我们查看make命令行。make命令行中有个make:message命令，可以帮助我们创建一个消息和一个消息处理器。

复制命令行，运行命令行。消息的名称我们叫做SendEmailMessage，命令行帮助我们创建了两个类。我们打开src目录，在Message目录中，命令行帮助我们创建了SendEmailMessage类，我们可以在类中添加任意的属性或者方法，然后将我们消息的内容设置到类中。我们打开Handler类，在__invode()方法中我们获取消息的内容，然后对消息的内容进行处理，我们使用SendEmailMessage来发送邮件，通知用户有新的文章需要处理，我们需要发送的文章的信息。

我们要使用消息发送文章信息，通常我们会定义一个属性，属性的名称叫做$post，使用$post属性来保存文章对象。但是我们的消息是要发送到传输或者队列上的，很可能需要通过网络传输，但是Post对象包含的信息非常多，通常如果我们发送的消息是个对象，我们使用对象的标识符进行发送，这里使用$postId。我们取消构造方法前的注释，参数这里输入$postId，下面也进行修改，取消get方法前的注释。进行修改，getPostId()，这里返回$postId。

```
#src/Message/SendEmailMessage.php

final class SendEmailMessage
{
    /*
     * Add whatever properties & methods you need to hold the
     * data for this message class.
     */

     private $postId;
//
     public function __construct(string $postId)
     {
         $this->postId = $postId;
     }
//
    public function getPostId(): string
    {
```

```
        return $this->postId;
    }
}
```

我们来修改Handler类，在__invoke()方法中我们从传输队列中获取到了消息，首先我们来获取消息中的$postId，通过$postId我们来获取数据库中的Post对象。

添加构造方法，在构造方法中，我们通过依赖注入的方式注入PostRepository对象，初始化属性，这里修改一下使用$this->postRepository，当我们获取到Post对象之后。我们来发送邮件，复制前面课程的订阅器代码，拷贝34行到40行的代码，粘贴，点击OK。

现在我们在构造方法中注入ParameterBag对象和Mailer对象，alt键加回车初始化属性，点击OK，现在就可以了，现在我们的Handler类就完成了。

```
#src/MessageHandler/SendEmailMessageHandler.php

final class SendEmailMessageHandler implements MessageHandlerInterface
{
    /**
     * @var PostRepository
     */
    private PostRepository $postRepository;
    /**
     * @var ParameterBagInterface
     */
    private ParameterBagInterface $parameterBag;
    /**
     * @var MailerInterface
     */
    private MailerInterface $mailer;

    public function __construct(PostRepository $postRepository,
ParameterBagInterface $parameterBag, MailerInterface $mailer)
    {
        $this->postRepository = $postRepository;
        $this->parameterBag = $parameterBag;
        $this->mailer = $mailer;
    }

    public function __invoke(SendEmailMessage $message)
    {
        $postId = $message->getPostId();
        $post = $this->postRepository->find($postId);

        $email = (new Email())
            ->from($this->parameterBag->get('send_email'))
            ->to($this->parameterBag->get('editor_email'), $this-
>parameterBag->get('checker_email'))
            ->subject('有新的文章<'.$post->getTitle().'>发布了，请检查。')
            ->text('有新的文章<'.$post->getTitle().'>发布了，请检查。');
        sleep(10);
```

```
            $this->mailer->send($email);
        }
    }
```

但是我们的消息并没有使用，我们需要将消息发送到传输中，然后使用Handler类来处理消息。在下节课，我们继续学习messenger组件。