我们搜索symfony flash message，我们打开symfony.com这个网站上的文档，我们点到Flash Messages这里。文档中说，我们可以在用户的Session中保存一些特殊的消息，这些特殊的消息叫做flash消息。flash消息被设计为它只会被使用一次，我们查看一下下面的代码。

这里是表单提交的代码，当表单提交之后，我们可以使用addFlash()方法添加一个flash消息。它有两个参数，第一个参数是flash消息的类型，第二个参数是flash消息的内容。然后我们往下看，在模板中，我们可以使用app的getflashs()方法来获取不同类型的消息，然后对消息进行显示。

```
#src/Controller/PostController.php

class PostController extends AbstractController
{
    // ...
    #[Route('/post/{id1}', name: 'post_show', methods: ['GET', 'POST'])]
    #[ParamConverter('post', options: ['id' => 'id1'])]
    public function show(Request $request, Post $post,
EntityManagerInterface $entityManager,
                         PaginatorInterface $paginator, CommentRepository
$commentRepository): Response
    {
        $commentForm = $this->createForm(CommentType::class);
        $commentForm->handleRequest($request);
        if ($commentForm->isSubmitted() && $commentForm->isValid()) {
            if ($commentForm->get('submit')->isClicked()) {
                /**@var Comment $data * */
                $data = $commentForm->getData();
                $data->setPost($post);
                $entityManager->persist($data);
                $entityManager->flush();
            }

            $this->addFlash('success', '您的评论已成功提交！');
        }

        $query = $commentRepository->getPaginationQuery($post);
        $pagination = $paginator->paginate(
            $query, /* query NOT result */
            $request->query->getInt('page', 1), /*page number*/
            10 /*limit per page*/
        );
        return $this->render('post/show.html.twig', [
            'post' => $post,
            'pagination' => $pagination,
            'comment_form' => $commentForm->createView()
        ]);
    }
    // ...
}
```

app对象是Symfony提供的一个全局对象，可以在Twig模板中使用这个全局对象。我们回到Controller，在文章详情页，当表单进行提交后，我们也添加一个flash消息。

使用$this->addFlash()方法，第一个参数是消息的类型，我们这里输入success。第二个参数是消息的内容，这里我们输入您的消息已成功提交。然后我们回到模板文件，在show.html.twig文件中，我们在content区块最顶部获取flash消息。

回到文档，我们复制最后一段代码，它会遍历app中的所有flash消息，在循环体中，使用div元素对消息进行了包裹，我们调整一下div的样式类，其中label就是消息的类型。这里我们输入alert，mt-4 alert

```
#templates/post/show.html.twig

{# read and display all flash messages #}
{% for label, messages in app.flashes %}
    {% for message in messages %}
        <div class="alert-{{ label }} mt-4 alert">
            {{ message }}
        </div>
    {% endfor %}
{% endfor %}
```

回到详情页刷新，我们手动提交一篇评论，点击提交。我们看到在文章的标题上面成功的显示了一个提醒。我们再次访问这个页面，现在这条提醒就已经没有了。

回看评论表单，在某些情况下，我们需要在表单中添加文件上传的功能。在下节课我们将创建一个文件类，用来保存表单提交的文件数据。