我们先来解决一下上节课遗留的问题，我们打开博客系统的首页，它提示错误了。在index.html.twig第39行path方法在生成post_show路由的链接时出错了。回到我们的项目，我们已经将路由的参数id修改为了id1，而这里传的参数还是id我们要修改一下。打开index.html.twig文件，这里将id修改为id1就可以了。

再次刷新，我们点击show链接。我们想在博客的详情页添加评论功能，评论功能，首先它有一个评论表单，当有用户通过评论表单提交数据时，我们就可以在管理端看到评论的内容，我们搜索symfony form，Symfony提供了form组件，可以让我们快速的根据Entity类来生成表单。首先我们要安装form组件。然后在命令行下方有一个简单的使用说明。

第一步首先是在controller方法中使用已经声明的Form类构建表单对象。然后是第二步将表单对象渲染到模板中，这样用户就可以编辑提交数据了。第三步就是处理表单的提交，然后将表单的数据转换为PHP数据，然后做一些操作，比如说保存到数据库。

继续向下看，Symfony有个表单类型的概念，比如说我们常用的文本输入框，这个表单行也是一个表单类型，Symfony使用了TextType类来对文本输入框进行了封装。你可能有很多表单行来保存用户的地址，它也是个表单类型。可以在Symfony中自定义创建一个表单类型的类，来对这些表单行进行封装，整个表单也可以用作表单类型。

Symfony内置了很多表单行的类型，我们点击进去，比如说文本类型，选择框类型，时间日期类型。我们可以将Entity类的各个属性指定一个表单行的类型，然后将所有的表单行封装成一个表单类型。

回到上一页下拉，这里有个示例代码，通过controller的createFromBuilder()方法返回一个FormBuilder对象。然后通过FormBuilder的add()方法，将Entity类的属性添加为表单行，最后通过getForm()方法来获取一个表单对象。

再往下看，我们还可以创建一个表单类，表单类继承于AbstractType。在类中使用buildForm()方法来构建表单，在方法中使用$builder的add()方法，将Entity类的各个属性添加为一个表单行。最后将所有的表单行封装成一个表单类型Type。

再往下看，在controller中使用render()方法将Form对象的View对象渲染到模板中，然后使用模板方法form()将form变量渲染出来。这样就完成了表单的显示。

我们回到项目，之前我们使用make:crud命令时创建了一个Form文件夹，在文件夹下就是文章的表单类型。我们打开这个类，buildForm()方法中使用Builder对象将文章的各个属性添加为表单行，在configureOptions()方法中设置表单的数据要转换为的类为Post类。然后我们在controller中可以直接使用表单类来创建form对象，再将form对象渲染到页面中，现在我们不需要文章类型的表单，删除这个类。

我们要使用make命令行创建评论表单，打开底部控制台，输入symfony console list make。这里有个make:form命令行，用来创建表单类，我们复制命令行，输入symfony console，粘贴命令行，创建表单类，表单类的类名我们叫做CommentType。

CommentType所对应的Model类的类名，我们这里输入Comment，在src/Form目录下，创建了CommentType类。我们查看这个类，命令行自动的将Comment类的各个属性添加到了表单中。我们回到Controller，在Controller中创建表单对象，我们在文章详情页创建表单对象，添加一个变量$commentForm等于，我们可以使用Controller的createForm()方法来创建表单对象，表单的类型就是我们刚刚创建的CommentType。这里输入类名 其他参数可以暂时忽略，然后我们将表单的View对象渲染到模板中。

```
class PostController extends AbstractController
{
    // ...

    #[Route('/post/{id1}', name: 'post_show', methods: ['GET'])]
    #[ParamConverter('post', options: ['id' => 'id1'])]
    public function show(Request $request, Post $post): Response
    {
        $commentForm = $this->createForm(CommentType::class);

        return $this->render('post/show.html.twig', [
            'post' => $post,
            'comment_form' => $commentForm->createView()
        ]);
    }

    // ...
}
```

打开模板文件，在模板文件中，我们使用form()方法来渲染表单，回到浏览器刷新文章详情页，这里提示文章对象不能转化为string类型，我们回看项目代码。打开CommentType类，在CommentType中添加了一个文章属性，我们不需要这个表单行，注释这一行代码，回到浏览器刷新。

在详情页下方就添加了一个表单，我们不需要createdAt、 updatedAt回到项目，注释这两行代码，刷新，我们表单并没有添加提交按钮，这样用户的数据就无法进行提交了。

```
#src/Form/CommentType.php
class CommentType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array
$options)
    {
        $builder
            ->add('author')
            ->add('email')
            ->add('message')
//            ->add('createdAt')
//            ->add('updatedAt')
//            ->add('post')
            ->add('submit', SubmitType::class)
        ;
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults([
            'data_class' => Comment::class,
        ]);
    }
```

```
}
```

在下节课，我们将有两种方式来添加提交按钮，并且处理表单的提交。