

我们打开底部控制台，输入`symfony console list make`，查看一下所有的make命令，这里有个`make:crud`命令行，它可以为doctrine的Entity类创建CRUD页面，我们复制这个命令行，输入`symfony console make:crud`。

这里出错了，它提示我们需要安装`validator annotations`组件，我们来复制这个命令行，执行，安装完之后。我们再次执行`make:crud`命令，这里提示我们要为哪个Entity类创建CRUD，我们选择`Post`，选择一个Controller类名，它这里提示我们使用`PostController`，我们直接回车命令行为我们创建了很多文件。

有`PostController`，还有`PostType`，然后是很多的Twig模板文件，最后命令行提示我们在`/post`路径下检查CRUD页面，确保我们的服务器正在运行，我们打开页面。输入`127.0.0.1:8000/post`，现在`/post`路径下显示了我们数据库中的所有文章，并且我们可以显示文章的详情，编辑文章，还可以创建新的文章

使用`make:crud`命令行，也为`Post`类创建了CRUD页面。我们已经有管理端页面了，所以我们并不需要这么多页面，我们需要进行一些删除，在`templates`中，我们打开`post`目录，我们只保留文章的显示和首页的列表显示，删除其他4个文件。

我们再来检查Controller类，打开`PostController`文件，命令行为我们创建了五个action方法，我们只需要`index()`方法和`show()`方法。`index()`方法用于文章列表的显示，`show()`方法用于显示文章的详情。我们删除剩下的3个方法。

我们看到`PostController`的类前有一个Route注解，注解中是一个`/post`路径，类前的注解是为所有action方法的注解添加路径前缀，我们删除类前的注解 我们希望`post_index`的这个路由的页面作为我们博客系统的首页。

然后我们修改`show()`方法，在路径前添加`/post`，在`index()`方法中，通过依赖注入传入了`PostRepository`类的对象。还记得我们之前课程中讲到的`handleRaw()`方法吗？我们再回顾一下，我们双击shift键，输入`HttpKernel`，如果你没有找到这个类的话，你需要勾选这里。

我们找到`handleRaw()`方法，在`handleRaw()`方法第149行，通过`$this->argumentResolver`的`getArguments()`方法，为Controller的action方法的参数设置值，最终我们就可以直接在action方法中，直接使用到依赖注入的服务类对象。

在下节课，我们将完善一下`index()`方法。