

在回复表单上点击右键检查一下表单代码，回复表单的action地址就是我们的`reply_comment`路由的地址。我们需要在`replyComment()`方法中，来处理表单的提交。

使用Form对象的`handleRequest()`方法来处理表单的请求，添加一个条件判断。如果回复表单已经提交了，`isSubmitted()`，并且回复表中的CSRFtoken是正确的。我们来获取表单的数据，输入一下`dd()`，查看一下表单的数据。

回到浏览器，我们手动的填写一下表单，表单提交的数据自动的转换为了Comment对象。Comment对象有作者、邮箱、消息。对于子级评论，我们可以不设置评论的post属性，但是我们要设置parent属性，用来设置父级评论对象。

回到代码，添加一个注释，`$data`数据是一个Comment对象，我们设置一下评论的父级对象，就是`$parentComment`。然后我们注入EntityManager对象，保存Comment数据，提交一下数据。保存完表单提交的数据之后，我们需要让页面跳转到文章的详情页。使用`redirectToRoute()`跳转到`post_show`路由，需要传递一个参数`id1`，参数的值就是Post对象的id。

```
#src/Controller/CommentController.php
class CommentController extends AbstractController
{
    #[
        Route('/post/{post_id}/comment/{comment_id}/reply', options:
        ['expose' => true], name: 'reply_comment'),
        ParamConverter('post', options: ['id' => 'post_id']),
        ParamConverter('parentComment', options: ['id' => 'comment_id']),
    ]
    public function replyComment(Request $request, Post $post, Comment
    $parentComment, EntityManagerInterface $em): Response
    {
        $replyComment = $this->createForm(CommentType::class, null, [
            'action' => $request->getUri()
        ]);

        $replyComment->handleRequest($request);

        if ($replyComment->isSubmitted() && $replyComment->isValid()){
            /**@var Comment $data**/
            $data = $replyComment->getData();
            $data->setParent($parentComment);
            $data->setPost($post);
            $em->persist($data);
            $em->flush();

            return $this->redirectToRoute('post_show', ['id1' => $post-
            >getId()]);
        }

        return $this->render('comment/_reply_comment_form.html.twig', [
            'reply_comment_form' => $replyComment->createView(),
        ]);
    }
}
```

回到浏览器，刷新再次提交表单，现在提示错误了，评论对象的文章id不能设置为空，那我们设置一下子评论的文章属性。刷新再次提交数据，现在子评论就成功的插入到了数据库中，但是我们提交的子评论并没有嵌套显示在父评论下方。

回到项目，我们打开`show.html.twig`文件，我们把之前嵌套显示的评论注释掉了，我们看注释的代码第71行，71行是条新的评论，它在父级评论的消息正文后面，我们复制71行到77行代码。拷贝，粘贴到回复按钮下方，取消代码的注释，我们打开`Comment`类。

在`Comment`类中，我们可以通过`getChildren()`方法来获取当前评论的所有子评论，回到`show.html.twig`文件，我们使用`comment`的`getChildren()`方法来获取所有子评论。

使用`for`循环来遍历所有子评论，`{% for childComment in comment.children %}`，结束子评论，然后我们修改子评论的作者和子评论的内容。然后在子评论的消息后面，我们也添加一个回复按钮，回复按钮的文章id不变，但是`data-parent-id`属性就是我们子评论的id属性。

但是现在还有一个问题，如果子评论它也有下级评论，该怎么办呢？我们还需要再复制代码吗？这样代码就无限的复制了。

回到浏览器打开Twig文档，twig.symfony.com，打开文档，Symfony为我们提供了一个`macro`关键字。我们查看`macro`关键字的文档，`macro`关键字可以定义一个方法，在方法中显示html代码，我们还可以在`macro`的方法中进行自身的递归调用。

回到项目代码，在`templates`目录中我们新添加一个`macro.html.twig`文件，在`macro.html.twig`文件中，我们可以定一个`macro`方法，方法名称叫做`show_comments()`。它是一个方法，需要一个括号，然后我们结束一下`macro`。

我们回到`show.html.twig`文件，我们首先获取文章的所有评论，作为父级评论，我们把`for`循环中的代码复制一下，粘贴到`macro`的`show_comments()`方法中。我们需要传入一个参数，在`show_comments()`方法中，我们用到了`comment`变量，这个变量需要我们通过参数来传入，我们还用到了`post`变量，也需要使用参数传入。在第8行，通过`for`循环的方式显示所有的子评论，我们可以在第8行的循环体中再次调用自身的`show_comments()`方法，实现一个递归调用。

在第二行前添加一行代码，我们使用`import`关键字，将`macro.html.twig`文件引入到`show_comments()`方法中，`_self`关键字代表`macro.html.twig`文件。引入后作为`macro`变量，在第9行的循环体中，我们删除代码。直接调用`macro`的`show_comments()`方法。

第一个参数我们传入当前子评论对象，第二个参数我们传入`post`对象，这样当在子循环中调用`show_comments()`方法时，它会传入子评论对象和文章对象再次显示评论数据。如果遇到评论数据还有子对象，那么将会再次递归调用`show_comments()`方法，我们回到`show.html.twig`文件，删除放循环中的代码。

```
#templates/marco.html.twig

{% macro show_comments(comment, post) %}
    {% import _self as macro %}
    <div class="media mb-4">
        
```

```
<div class="media-body">
    <h5 class="mt-0">{{ comment.author }}</h5>
    {{ comment.message }}
    <button class="btn btn-sm btn-link js-reply-comment-btn" data-
post-id="{{ post.id }}" data-parent-id="{{ comment.id }}">回复</button>
    {% for childComment in comment.children %}
        {{ macro.show_comments(childComment, post) }}
    {% endfor %}
</div>
</div>
{% endmacro %}
```

首先我们在文件顶部引入`macro.html.twig`文件, 然后作为变量`macros`。在评论的for循环体中, 我们使用`macros`变量的`show_comments()`方法来递归的显示评论列表。第一个参数我们传入`comment`, 第二个参数我们传入`post`。

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

{% extends 'base.html.twig' %}
{% import 'macro.html.twig' as macros %}

<!-- Single Comment -->
{% for comment in post.comments %}
    {{ macros.show_comments(comment, post) }}
{% endfor %}
```

回到文章详情页刷新, 现在子评论就按照递归的方式显示了, 我们再添加一条评论。提交, 再次查看评论列表, 出现了问题。我们新添加的评论已经显示在对应的位置了, 但是在评论的列表前, 它又进行了额外的显示。这是因为所有的子评论对象, 它们也都设置了`post`属性, 那么在`post`获取所有评论对象时, 将所有的子评论当做顶级评论对象进行了遍历。

另外我们点击回复按钮, 添加了一个回复评论框, 当再次点击时, 在评论框下方又多了一条评论框, 我们想让评论表单只显示一次。

在下节课我们将解决这些问题。