

在文章详情页, 当我们点击回复按钮时, 会发送一个Ajax请求。将父评论的id和文章的id当做参数传递到一个controller方法中, 然后在controller方法中动态的生成一个评论表单, 然后再使用jquery方法将表单动态的添加到评论消息的下方。

我们先来完成这个controller方法。回到项目, 打开控制台, 我们输入`symfony console make:controller`, 来创建一个Controller, Controller的类名, 我们叫做`CommentController`。

打开`CommentController`类, 修改一下路由的路径, 我们要生成一个回复评论表单, 传递的参数是文章的id和父评论的id, 我们可以把这两个id添加到路由的参数中, 路由路径我们修改为`/post/{post_id}/comment/{comment_id}/reply`, 修改方法名称为`replyComment()`。

通过传入的参数, 我们可以直接获取到文章对象和父评论对象, 通过依赖注入的方式直接使用, 引用一下`Post`类和`Comment`类。占位符的名称和对象id并不相同, 我们需要使用`ParamConverter`注解来设置一下, 第一个参数我们先设置`post`, 参数名就是`$post`, 然后是`options`。

我们让`id`属性指向`post_id`, 再来设置父评论对象, 按着`command`键+d复制一行, 属性名就是`parentComment`, 然后同样的id我们指向`comment_id` 这样就设置好了。

在controller方法中, 我们可以使用Controller提供的`createForm()`方法来创建表单, 表单的类型就是我们之前创建的`CommentType`类型, 这里直接使用`CommentType`全类名。

它的返回值我们设置一个变量, `createForm()`有三个参数, 第一个参数就是表单的类型。第二个参数是表单的数据, 是在数据回显时可以传入第二个参数, 通常在编辑表单中需要传入第二个参数。第三个参数就是对表单的一些设置。

我们需要修改回复表单的提交地址, 提交地址就设置为我们当前的路由地址, 第二个参数我们传入`null`, 第三个参数我们设置一个数组, 设置`action`就是我们当前的路由路径。

我们需要注入Request对象, 使用`HttpFoundation`下的Request对象, 我们通过`$request->getUri()`来获取当前的路由地址, 然后我们创建一个模板文件, 将表单渲染到模板文件中。

在`templates`目录中我们打开`comment`文件夹, 删除`index.html.twig`文件。我们创建一个Twig文件, 名称叫做`_reply_comment_form.html.twig`文件。我们在`render()`方法中将表单的View对象传入模板文件, 这里输入`reply_comment_form`, 修改模板名称。

```
class CommentController extends AbstractController
{
    #[
        Route('/post/{post_id}/comment/{comment_id}/reply', name:
        'comment'),
        ParamConverter('post', options: ['id' => 'post_id']),
        ParamConverter('parentComment', options: ['id' => 'comment_id']),
    ]
    public function replyComment(Request $request, Post $post, Comment
    $parentComment): Response
    {
        $replyComment = $this->createForm(CommentType::class, null, [
            'action' => $request->getUri()
        ]);
        return $this->render('comment/_reply_comment_form.html.twig', [
```

```
        'reply_comment_form' => $replyComment->createView(),
    ]);
}
}
```

再回到模板文件使用 `form()` 方法来渲染表单，表单名称就是 `reply_comment_form`。回到浏览器，我们手动的输入回复表单的路由地址，查看一下最新的评论id，id是4，它成功的返回了一个表单。我们可以在文章详情页使用jquery方法动态的访问这个路由，然后生成表单，再将表单渲染到评论的下方。

回到项目，我们打开assets目录下的app.js文件，这里我们调整一下jquery和bootstrap的顺序。当页面文档加载完成后，我们在jquery的ready()方法中处理回复按钮的事件，获取回复按钮的DOM元素。

回到show.html.twig文件复制按钮的样式类，监听click事件，在click事件中当按钮点击时，我们向路由的地址发送两个参数，一个是父评论的id还有一个是文章的id。这里我们添加一下data-post-id属性，属性的值就是 `post.id`。在按钮事件中我们获取这两个id，首先是postId我们让他等于 `$(this).data('post-id')`。然后是parentId， `$(this).data('parent-id')`。

获取到这两个参数后，我们通过jquery的ajax()方法来发送一个请求，请求的地址我们这里先设置为空，然后是请求的类型type，我们使用POST类型。当请求完成后，我们使用done()方法来处理结果，如果请求发送失败，我们使用fail()方法来处理返回的结果。

```
#assets/app.js

import './bootstrap';
import $ from 'jquery';
import 'bootstrap';
$(document).ready(function(){
    $('button.js-reply-comment-btn').on('click', function () {
        let postId = $(this).data('post-id');
        let parentId = $(this).data('parent-id');

        $.ajax({
            url: '',
            type: 'POST'
        }).done(function (response){

        }).fail(function (jqXHR){

        })
    })
})
```

请求的地址我们可以使用硬编码的方式进行字符串的拼接，但是这样的话不够灵活。回到浏览器，我们搜索 `symfony FOSJSRouting`，`FOSJSRoutingBundle`可以在JS代码中动态的生成路由的路径。

在下节课，我们将使用 `FOSJSRoutingBundle`动态的生成路由的链接地址。