

我们打开Comment类，我们可以在Comment类中添加一个属性，用来控制评论的层级。默认情况下评论都是第一级，如果评论是子评论，那么子评论就要设置为父评论的层级再加1。

我们打开控制台，输入`symfony console make:entity`，来对Comment类进行修改，类名输入Comment，属性名输入`$level`，类型我们输入整数型，它在数据库中可以为空吗？不允许为空。直接回车退出命令行。

我们查看命令行的更改，在Comment类中添加了一个`$level`属性，我们让`$level`属性有个默认值等于1，这样如果是在文章下面提交的评论，它默认层级就是1。

```
#src/Entity/Comment.php

class Comment
{
    // ...

    /**
     * @ORM\Column(type="integer")
     */
    private $level = 1;

    //...

    public function getLevel(): ?int
    {
        return $this->level;
    }

    public function setLevel(int $level): self
    {
        $this->level = $level;

        return $this;
    }
}
```

如果是在评论列表回复评论，那么我们需要在controller方法中手动的设置它的层级。打开CommentController类，在表单提交这里我们设置一下，设置它的`setLevel()`等于`$parentComment->getLevel()+1`。我们再次修改controller方法，在controller方法中添加一个条件判断，我们定义一个变量来限制最大的回复层级。如果父评论的层级已经是第5级了，那么我们就直接返回一个提示，当前层级已经不允许用户提交评论了。

我们打开控制台，来创建一个数据库的更改，我们查看migration文件。comment表中添加了一个level列，它是INT类型，我们执行数据库的更改，yes。

回到浏览器，我们的手动测试一下代码。刷新详情页，我们先在文章下方的评论框中创建一个顶级评论。作者这里输入hello1，email，hello1，然后信息也是hello1，提交。我们在hello1下面点击回复按钮，作者hello2，email也是hello2，信息也是hello2，提交。再回复hello2，输入hello3。回复hello3，输入hello4我们再回复hello4，再次提交。

现在我们的子评论hello5已经达到了限制层级，我们点击回复按钮，其实我们已经达到最大层级，不允许再添加子评论了。我们再次点击回复，出现了一个小bug，我们通过修改JS代码来修改这个bug。检查一下限制提示，在回复按钮后面只添加了两行文字，我们需要使用标签来包裹这两行文字，修改 `CommentController`。

在文字前我们添加html标签，标签类我们输入 `max-level-info`，打开assets下的app.js。我们添加判断，如果 `$(this)`，button按钮的后面的p标签的长度为1，我们直接结束按钮事件的处理。

```
#assets/app.js

$('button.js-reply-comment-btn').on('click', function (element) {
    let postId = $(this).data('post-id');
    let parentId = $(this).data('parent-id');
    if ($(this).nextAll('p.max-level-info').length === 1){
        return;
    }
    if($(this).nextAll('div.reply-comment-card').length === 0){
        $.ajax({
            url: Routing.generate('reply_comment', {post_id: postId,
comment_id: parentId}),
            type: 'POST'
        }).done(function (response){
            $(element.target).after(response)
        }).fail(function (jqXHR){
        })
    }
})
```

回到浏览器，我们刷新，我们点击hello5后面的回复按钮，再次点击，现在这个bug就已经修复了。回到项目，我们可以在配置文件中设置最大的限制层级，打开config目录下services.yaml文件，添加一个参数，设置为5。

```
#config/services.yaml

parameters:
    # ...
    max_comment_level: 5
```

然后回到Controller,我们可以使用Controller的 `$this->getParameter()` 来获取这个参数。

```
#src/Controller/CommentController.php

public function replyComment(Request $request, Post $post, Comment
$parentComment, EntityManagerInterface $em): Response
{
    $maxLevel = $this->getParameter('max_comment_level');
```

```
        if ($parentComment->getLevel()==$maxLevel){
            return new Response('<p class="max-level-info">当前评论已达到最大层级,
不允许添加子评论。😞 </p>');
        }

        $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);
            $data->setLevel($parentComment->getLevel() + 1);
            //          $data->setPost($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(),
        ]);
    }
```

我们已经手动的测试了子级评论的提交，我们希望程序能够自动化的进行测试。在下节课，我们将开始使用代码的方式来进行子评论的提交。