

打开CommentCrudController类,我们查看一下`configureFields()`方法。在`configureFields()`方法中定义了一些字段,这些字段的设置就可以控制管理端列表各个列的显示。

回到浏览器,我们想在列表上添加一个图像的显示列,我们就需要在`configureFields()`方法中添加一个图片显示的字段配置。我们需要自定义一个字段类来显示图像列。

回到浏览器,我们搜索EasyAdminBundle,打开`symfony.com`这个网站。打开Fields文档,在Fields文档这里有创建自定义字段的文档。点击,这里有段示例代码,自定义的字段需要实现`FieldInterface`,然后在`FieldInterface`的`new()`方法中,我们可以对字段进行一些设置。比如说设置属性,设置标签,设置模板,然后添加资源文件。再到`configureFields()`方法中直接使用自定义的字段。

回到代码,我们在src目录中新增一个目录,目录的名称叫做Field。我们新增一个类,类名叫做`CommentImageField`。`CommentImageField`要实现`FieldInterface`,我们选择EasyAdminBundle的`FieldInterface`,实现`FieldInterface`的方法。鼠标移动到红线上,点击添加方法,点击OK。

回到文档,我们复制一下代码,在`new()`方法中我们直接返回一个自身对象,然后对自身对象进行一些设置,自身对象现在只有`getAsDto()`方法。我们可以使用`FieldTrait`来扩展类的方法,使用`use FieldTrait;`。

我们可以使用`setProperty()`来设置字段的属性,属性名称就是参数`$propertyName`,然后我们可以设置label,标签就是参数`$label`。我们还可以设置首页列的模板,需要指定一个模板路径,我们在`templates`目录下创建一个`fields`目录,在`fields`目录下新增一个模板文件,叫做`comment_images.html.twig`。

回到CommentImageField类,路径就是`fields/comment_images.html.twig`文件,这个模板文件用于在列表中列的显示。我们查看一下`FieldTrait`源码。查看一下`FieldTrait`的结构,在`FieldTrait`方法中已经有了`getAsDto()`方法,我们可以删除CommentImageField中的`getAsDto()`方法。

```
#src/Field/CommentImageField.php

class CommentImageField implements FieldInterface
{
    use FieldTrait;

    public static function new(string $propertyName, ?string $label = null)
    {
        return (new self())
            ->setProperty($propertyName)
            ->setLabel($label)
            ->setTemplatePath('fields/comment_images.html.twig');
    }
}
```

回到CommentCrudController,我们在`configureFields()`方法中使用我们自定义的字段,在评论消息后面我们添加一列, `CommentImageField::new()`属性名称使用(口误)`files`。

现在EasyAdminBundle会将一些数据渲染到我们的模板文件,我们查看一下模板中可以用的变量。回到浏览器刷新评论列表,现在图片的列就显示了,模板中有四个变量。我们查看一下`field`变量, `field`变量有个`value`

属性。我们查看value属性，value属性是个集合。我们查看一下collection属性，当前评论没有图像。如果有图像的话，我们可以直接遍历field变量的value属性。

回到项目，我们添加循环遍历`{% for file in field.value %}`，结束循环`{% endfor %}`。在循环体中，我们可以遍历显示所有的缩略图片，我们复制之前编写的代码，在macro.html.twig文件中，我们复制之前编写`lightbox2`代码。回到模板文件，粘贴，删除dump()方法。

回到浏览器刷新，我们查看最后一页，这里参数出错了，现在没有comment变量。回到代码，我们再次查看dump()方法，刷新。模板的entity变量中，我们可以获取到当前的评论对象，在entity变量的instance属性中，我们查找到了当前的comment对象。回到模板，我们输入`entity.instance.id`，就是当前评论的id。注释dump()方法。

```
#templates/fields/comment_images.html.twig

{#{ dump() }#}
{% for file in field.value %}
    <a href="{{ asset(file.path) }}" data-lightbox="{{ entity.instance.id }}">
        
    </a>
{% endfor %}
```

回到浏览器刷新，我们看最后两个评论，显示了图像的缩略图，我们可以在管理端引入lightbox2 JS库。我们查看EasyAdminBundle的文档，打开文档首页，我们查看Design设计文档往下看，在CrudController中有个`configureAssets()`方法，可以在Assets对象中添加CSS文件或者JS文件。

回到CommentCrudController类，我们覆盖方法，为Assets对象添加JS文件，再添加CSS文件，我们需要传入文件的路径。将我们之前下载的lightbox2 JS库复制到public目录中。找到lightbox2文件夹，我们复制dist文件夹，拷贝，回到public目录。在public目录中新建一个文件夹叫lightbox2，粘贴。

在configureAssets()方法中，我们添加JS文件，lightbox2依赖于jquery，我们复制lightbox-plus-jquery.js文件，拷贝路径，粘贴，删除public前缀。复制CSS文件，拷贝，粘贴，删除public前缀。

回到浏览器刷新评论列表，我们再次点击缩略图，现在缩略图就弹出了一张图片，我们还要调整一下管理端评论列表的显示顺序。我们让id按照降序的方式进行排列，回到代码，我们覆盖`configureCrud()`方法。通过Crud对象的setDefaultSort()方法，我们设置id按照降序的方式进行排列。回到浏览器刷新。

我们看第一页，现在id就按照降序的方式进行了显示，我们只想让图片在首页上显示，另外我不需要在首页上显示更新日期。回到代码，在files列上我们添加`onlyOnIndex()`方法。我们删除更新日期这一列，回到浏览器刷新，这样就可以了。

我们想在评论列表上增加一列显示评论所属于的文章，回到项目评论类和文章类之间存在着关联关系。EasyAdminBundle提供了很多内置的字段类，我们随便查看一个内置的字段类，按着command键点击TextField。我们打开TextField所在的目录，在Field目录中我们看到了一个跟关联关系有关的类。

回到CommentCrudController，在message列后面添加一列，我们使用这个类，属性我们使用post属性。回到评论列表，刷新，现在评论所属于的文章已经显示了。但是文章显示的是id，回到项目，我们打开Post

类, 在Post类中, 我们增加`__toString()`方法, 返回文章的标题, 再次刷新页面。

现在文章列就显示了文章的标题, 但是子评论并不会显示文章的标题, 在提交子评论的Controller方法中, 子评论对象并没有设置post属性, 所以子评论不会显示文章标题。我们可以使用doctrine的事件在加载子评论时获取父级评论的文章对象, 然后进行标题的显示。

在下节课我们来了解一下doctrine的事件。