

在《Symfony 5全面开发》课程中, 我们使用`make:entity`命令行快速的创建了很多Entity类, 并且命令行自动为我们添加了ORM注解, 注解有2种, 一种是写注释里的, 一种是php8.0新增加的原生的注解。php8.0的注解在Symfony的新版本里也得到了支持。

但是如果你开发的Bundle是面向更低版本的Symfony, 那么还是推荐使用xml文件的方式这样兼容性更好。如果你的Bundle是用在最新版本 Symfony 项目中那么推荐使用注解。注解更简单, 代码阅读起来更好理解。

我们使用一个懒方法, 使用make命令创建一个新的File类, 然后把File类移动到我们的Bundle目录中。新的类名叫做`FileAnnotation`:

```
<?php

namespace Teebb\UploadBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="teebb_file_annotation")
 */
class FileAnnotation
{
    /**
     * @ORM\Id
     * @ORM\GeneratedValue
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $originName;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $name;

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

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $mimeType;

    public function getId(): ?int
    {

```

```
        return $this->id;
    }

    public function getOriginName(): ?string
    {
        return $this->originName;
    }

    public function setOriginName(string $originName): self
    {
        $this->originName = $originName;

        return $this;
    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName(string $name): self
    {
        $this->name = $name;

        return $this;
    }

    public function getSize(): ?int
    {
        return $this->size;
    }

    public function setSize(int $size): self
    {
        $this->size = $size;

        return $this;
    }

    public function getMimeType(): ?string
    {
        return $this->mimeType;
    }

    public function setMimeType(string $mimeType): self
    {
        $this->mimeType = $mimeType;

        return $this;
    }
}
```

同样的, 我们需要让Doctrine读取FileAnnotation类的注解, 修改TeebbUploadBundle, DoctrineOrmMappingsPass提供了另一个方法用于读取注解, 我们现在的注解是注释里的Annotation类型, 使用createAnnotationMappingDriver。如果是php 8自带的Attribute注解, 使用createAttributeMappingDriver方法。

```
class TeebbUploadBundle extends Bundle
{
    public function build(ContainerBuilder $container)
    {
        parent::build($container);

        $mappings = [
            realpath(__DIR__ . '/Resources/config/doctrine-mapping') =>
            'Teebb\UploadBundle\Entity',
        ];
        $namespaces = [
            'Teebb\UploadBundle\Entity'
        ];
        $directories = [
            realpath(__DIR__ . '/Entity')
        ];

        if (class_exists(DoctrineOrmMappingsPass::class)) {
            $container->addCompilerPass(DoctrineOrmMappingsPass::createXmlMappingDriver($mappings));
            $container->addCompilerPass(DoctrineOrmMappingsPass::createAnnotationMappingDriver($namespaces, $directories));
        }
    }
}
```

最后在项目中使用命令行symfony console doctrine:mapping:info查看我们的类是否已经映射成功了。

Found 5 mapped entities:

```
[OK] App\Entity\Comment
[OK] App\Entity\Post
[OK] App\Entity\User
[OK] App\Entity\FileManaged
[OK] Teebb\UploadBundle\Entity\File
```

我们可以使用命令行创建表了。symfony console make:migration查看migration文件, 我们的映射已经成功起作用了。

现在我们的FileAnnotation类就可以使用了, 但是现在遇到了另一个问题。`FileAnnotation`的ORM映射我们写死了, 如果想扩展`FileAnnotation`增加一些其他属性, 我们可以修改Doctrine注解, 用来扩展`FileAnnotation`类。下节课, 我们补充一些Doctrine注解的额外知识。