

当我们的博客系统开发好之后,我可能会用这个博客系统来搭建自己的技术博客。当有读者按照技术博客的文章进行测试时,可能会出现问题。那么读者就想在评论中上传几张图片或者文件来和我交流,我们需要在评论表单中添加文件上传的功能,我们不限文件上传的数量,可以在Symfony中添加一个Model类来保存用户上传的文件数据。同时,文件对象和评论对象之间存在着一定的关联关系,我们首先来创建这个文件类。

回到项目,打开控制台,输入`symfony console make:entity`,类名我们叫做`FileManaged`。

现在我们添加类的属性,首先我们要保存一下文件的原始名称, `originName`, 名称我们使用string类型, 长度255, 不能为空。

然后我们保存一下文件的新名称, `fileName`, string类型, 255, 不能为空。

然后我们保存一下文件的大小, `fileSize`, 输入整数类型, 不能为空。

然后我们需要保存文件的类型, `mimeType`, 同样也是string类型, 长度255, 不能为空。

最后我们保存一下文件在服务器上的路径, 这样可以快速的使用文件, 输入`path`, string类型255不能为空。

回车, 现在我们就创建好了文件类, 我们来创建一个数据库的migration文件, 粘贴命令行, 查看一下migration文件。migration文件在数据库中创建了一个`file_managed`表, 我们想修改一下这个表的表名, 我们使用`file`作为表的表名。修改`FileManaged`类, 在类前的注释中, 我们添加一个注解`@ORM\Table`, 我们修改`Table`的表名为`file`。

```
#src/Entity/FileManaged.php

/**
 * @ORM\Entity(repositoryClass=FileManagedRepository::class)
 * @ORM\Table(name="file")
 */
class FileManaged
{
    /**
     * @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 $fileName;

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

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

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

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 getFileName(): ?string
{
    return $this->fileName;
}

public function setFileName(string $fileName): self
{
    $this->fileName = $fileName;

    return $this;
}

public function getFileSize(): ?int
{
    return $this->fileSize;
}

public function setFileSize(int $fileSize): self
{
    $this->fileSize = $fileSize;

    return $this;
}

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

```
}

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

    return $this;
}

public function getPath(): ?string
{
    return $this->path;
}

public function setPath(string $path): self
{
    $this->path = $path;

    return $this;
}
}
```

删除之前的migration的文件，重新创建migration文件，打开新的migration文件，现在会在数据库中创建一个新的表，表名就是file。

SQL语句现在没有其他问题，我们执行数据库的更改，输入yes。同样的我们在测试环境下也执行一下更改，yes。

我们并不限制评论表单中文件的数量，所以文件对象和Comment对象之间存在着关联关系，每条评论可以有多个文件，但是文件对象并不需要和评论对象之间建立关系，也就是说，我们并不需要通过文件对象来获取这个文件对象所属于的评论对象。这就是**单向的一对多关系**。

在下节课我们将修改Comment类，使用添加第三张表的方式来实现单向的一对多关系。