

我们搜索doctrine onetomany, 我们打开doctrine文档, 文档中这里有使用第三张表来实现单向一对多的案例。我们查看一下文档, 点击。往下看这里有段示例代码, 分别是用户类和手机号码类。用户有个手机号码属性, 每个用户可以有多个手机号码, 但是手机号码类中并没有用户属性。

在\$phonenumbers属性上使用了ManyToOne多对多关联关系, 但是在第10行添加的第三张表中, 我们看到手机号码id这一列, unique设置设置为true, 这样手机号码id这一列保存的数据就是唯一的。这样就通过了多对多的关联关系, 实现了一个单向的一对多。

回到项目, 打开控制台, 输入symfony console make:entity, 来对Comment类进行修改。新的属性我们叫做files, 字段类型这里我们使用ManyToOne, 指向FileManaged类, 它提示我们需要在FileManaged类中添加一个新的属性吗? 我们不需要。

回车退出命令行, 我们查看一下Comment类的更改, 在Comment类中添加了一个\$files属性, 然后添加个注解ManyToOne。我们回到文档, 复制示例代码上的注解, 粘贴, 修改一下注解。这里也需要修改, 表名这里我们叫做comments_files。

单向一对多, 一这一端的列名这里我们输入comment_id。它指向的是Comment类的id, 然后我们修改多这一端的列名为file_id, 它指向的是FileManaged类的id, 然后file_id, unique值设置为true。

```
#src/Entity/Comment.php

/**
 * @ORM\Entity(repositoryClass=CommentRepository::class)
 */
class Comment
{
    // ...

    /**
     * @ORM\ManyToOne(targetEntity=FileManaged::class)
     * @ORM\JoinTable(name="comments_files",
     *     joinColumns={@ORM\JoinColumn(name="comment_id",
     referencedColumnName="id")},
     *     inverseJoinColumns={@ORM\JoinColumn(name="file_id",
     referencedColumnName="id", unique=true)}
     * )
     */
    private $files;

    public function __construct()
    {
        $this->children = new ArrayCollection();
        $this->files = new ArrayCollection();
    }

    // ...

    /**
     * @return Collection|FileManaged[]
     */
    public function getFiles(): Collection
```

```
{
    return $this->files;
}

public function addFile(FileManaged $file): self
{
    if (!$this->files->contains($file)) {
        $this->files[] = $file;
    }

    return $this;
}

public function removeFile(FileManaged $file): self
{
    $this->files->removeElement($file);

    return $this;
}
}
```

我们打开控制台，创建一个migration文件，复制命令行粘贴。我们查看一下migration文件，migration文件创建了第三张表`comments_files`表。然后添加了两个外键约束，`comment_id`它引用的是comment表的id，然后`file_id`它引用的是file表的id，然后`file_id`它添加了唯一约束。

我们执行数据库的迁移，复制命令行粘贴，输入yes，同样的在测试环境下执行数据库迁移，我们打开数据库客户端，查看一下数据库表。刷新，现在数据库有个文件表，然后添加了评论文件表，然后第三张表的`file_id`它添加了唯一约束。

在下节课我们将修改Comment表单类型。