之前的课程我们已经使用Fixtures创建了一些假的文章数据，我们查看一下之前编写的代码。打开PostFixtures类，在load()方法中，我们创建了一些简单的文章数据后面用序号来做的区分。但是这些数据太简单了，我们想创建一些真实的文章数据。

打开浏览器搜索fakerphp，打开第一个页面。我们可以使用Factory的create()方法来创建一个$faker对象。然后使用$faker对象的方法来随机的生成姓名，邮件地址或者长文本等等不同类型的数据。我们复制命令行来安装一下fakerphp。

```
composer require fakerphp/faker
```

回到项目，打开控制台粘贴。回到PostFixtures类中，我们创建一个私有属性叫$faker，在构造方法中我们为$faker对象赋值。使用Factory的create()方法来创建$faker，这里选择Faker下的Factory。

我们修改一下for循环的数量，我们让它循环20次。然后修改一下文章的标题和文章的正文。文章的标题我们使用$faker对象来生成。$this->faker使用$faker对象提供的sentence()方法来生成一个句子用作标题。文章的正文，我们使用$faker对象的另外一个方法，$this->faker->paragraph()，来生成段落。

文章的发布状态这里，之前我们设置为偶数时，它为已发布状态。现在我们使用$faker的另一个方法，输入$this->faker->boolean()方法 它会随机的生成一个布尔值，然后布尔值为真的话设置为已发布的状态。

我们还想设置文章的封面图像，我已经在项目的course目录下添加了一些假的图片，我们把这些假的图片拷贝一下。我们放置到public下的uploads/images目录下。粘贴。

在load()方法中，我们随机的指定一下封面图像的文件名。添加一个变量$image='00'，我们让$faker对象随机的在0到9之间生成一个数字，这里输入randomDigit()。然后再加上文件的后缀，我们设置一下文章的封面图像。

我们想在最新的一篇文章上添加一些假的评论数据，我们对第20篇文章添加一些设置，如果$i等于19，那么我们让$post设置为已发布状态。

```
#src/DataFixtures/PostFixtures.php

class PostFixtures extends Fixture
{
    /**
     * @var PostFactory
     */
    private $postFactory;
    private $faker;

    public function __construct(PostFactory $postFactory)
    {
        $this->postFactory = $postFactory;
        $this->faker=Factory::create();
    }

    public function load(ObjectManager $manager)
```

```
    {
        for ($i = 0; $i < 20; $i++) {
            $post = $this->postFactory->create($this->faker->sentence(),
$this->faker->paragraph());
            if ($this->faker->boolean()){
                $post->setStatus('published');
            }

            $image = '00'.$this->faker->randomDigit().'.jpg';
            $post->setPostImage($image);

            if ($i == 19){
                $post->setStatus('published');
            }

            $manager->persist($post);
        }
        $manager->flush();
    }
}
```

在下节课，我们将创建一些假的评论数据添加到最后一篇文章上，我们将学习到如何在Fixtures中实现数据的共享。