我们打开浏览器搜索doctrine fixtures bundle，我们打开第一个symfony.com这个页面上的文档，文档中说Fixtures用于加载一些假的数据到数据库中。这样你在测试时或者开发时可以提供一些有趣的数据，它的安装方法就是使用composer命令行。

```
composer require --dev orm-fixtures
```

我们复制这个命令行回到项目中，打开底部控制台粘贴命令行，现在我们查看一下make命令，make命令中有一行make:fixtures。就可以帮助我们创建一个Fixtures类，我们通过这个类可以在数据库中创建一些假的数据，复制命令行。输入symfony console，粘贴命令行。

我们想在数据库中创建一些假的文章数据，类名我们输入PostFixtures，它在src目录下的DataFixtures目录下，创建了PostFixtures类，打开这个类。在load()方法中，它们提供了两行示例代码，它首先new一个对象，再使用$manager的persist()方法，将new的对象保存到数据库中。

我们想创建Post对象，还记得我们的PostFactory服务类吗，我们可以在PostFixtures的构造函数中直接使用PostFactory对象。下划线下划线，构造函数的参数就是PostFactory，按着alt键加回车，我们初始化属性。这样我们就可以在load()方法中直接使用$postFactory属性，这就是在服务类中依赖注入的使用。

下面我们可以直接使用$postfactory了，我们通过for循环的方式来创建文章。在循环体中，我们使用PostFactory的create()的方法来创建Post对象。title我们这里输入，后面添加序号。然后正文。我们让$i为奇数时，它的发布状态是草稿状态，偶数时它的发布状态为已发布状态。如果$i模上2等于0，那么它的发布状态$post，发布状态为published，然后我们使用$manager->persist()方法，保存Post对象。在循环体外使用flush()方法提交数据。

```
#src/DataFixtures/PostFixtures.php
class PostFixtures extends Fixture
{
    /**
     * @var PostFactory
     */
    private $postFactory;

    public function __construct(PostFactory $postFactory)
    {
        $this->postFactory = $postFactory;
    }

    public function load(ObjectManager $manager)
    {
        for ($i = 0; $i < 5; $i++) {
            $post = $this->postFactory->create('Fake Post Title ' . $i,
'Fake Post Body ' . $i);
            if ($i %2 ==0){
                $post->setStatus('published');
            }

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

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

我们打开底部控制台，在命令行的下方提示我们，使用命令行将fixtures中的数据加载到数据库中，我们复制命令行，粘贴，回车。使用`doctrine:fixtures:load`命令行时，数据库将会被清除，我们要继续吗？是的。现在我们Fixtures中的数据就已经保存到数据库了。

我们打开数据库客户端，我们打开teebblog数据库，我们查看post表中的数据，我们的数据已经成功插入到了表中，如果我们想在开发环境下加载数据怎么办呢？还记得`--env`选项嘛，我们把`--env`选项设置为test命令，再次运行，yes。

现在我们查看测试数据库中的数据，我们数据同样插入成功了，fixtures-bundle用来对项目的数据进行初始化时非常有用。比如说管理员账户的创建、一些基本设置的设定等等。

到本节课我们已经学习了测试驱动开发，还有doctrine的基本使用，在下节课我们将安装我们博客系统的管理端。