打开之前编写的功能测试代码，我们打开PostDetailTest类，在测试方法中首先创建了一个Client对象。我们按着command键鼠标点击createClient()方法。createClient()方法，最终的返回对象是个KernelBrowser对象。

按着command键再次点击KernelBrowser类，进入KernelBrowser类它继承自HttpKernelBrowser，我们再次进入HttpKernelBrowser。HttpKernelBrowser它继承了AbstractBrowser，我们进入AbstractBrowser，AbstractBrowser它是browser-kit的一个类。

browser-kit它是Symfony的一个组件，我们之前编写的功能测试代码，都使用了browser-kit的内置浏览器来发送请求进行测试，但是它并不是一个真正的浏览器，并不能运行JS代码。我们需要使用一个真正的浏览器来进行功能测试。

打开底部控制台，新建控制台，输入symfony console make:test，最后一个类型PantherTestCase。它会使用一个真正的浏览器来进行功能测试。

我们打开浏览器搜索symfony panther。我们打开github这个页面，向下查看一下README文件。Panther是一个方便的独立库，用于抓取网站并使用真实的浏览器进行端对端的测试。

我们来安装一下Panther，复制命令行，我们将Panther安装到开发依赖下，回到项目新建一个终端粘贴命令。命令行会提示我们安装chrome浏览器的驱动或者火狐浏览器的驱动，我们复制第一个命令行。

这个命令行会在项目中安装一个浏览器驱动程序的安装文件，然后使用第二个命令行来检测并安装浏览器驱动，命令行会自动的在项目的drivers目录下安装对应浏览器的驱动。我们回到浏览器查看README文档，你也可以在不同的系统中使用命令行来安装浏览器驱动。

回到项目，我们在看命令行第3个提示，取消phpunit.xml.dist文件中的注释。我们打开项目根目录下的phpunit.xml.dist文件。往下拉，在第36行有个注释。

我们启用phpunit的Panther扩展，现在Panther的测试环境就已经准备好了。

在下节课，我们将编写代码来使用真正的浏览器，来测试回复评论的提交。