

我们搜索`symfony security`, 查看Symfony的安全文档, 往下看。这里有段是关于用户是否已经登录的文档, 我们点击[这里](#), 如果你只想检查用户是否已经登录了, 并不关心用户的角色。我们有两种方法, 第一种方法是检查`ROLE_USER`这个角色, 因为所有的用户都有`ROLE_USER`角色。

另外一种方法可以使用一个特殊的属性, 叫作`IS_AUTHENTICATED_FULLY`。这个属性它不是一个角色, 它更像是一种行为。所有(使用账号密码)已经登录过的用户都有这个属性, 我们可以通过检查这个属性来判断用户是否已经登录。

另外还有一些其他的属性, 比如说`IS_AUTHENTICATED_REMEMBERED`, 这个属性是通过勾选记住我(Remember me)选项, 自动登录的用户会有这个属性, 通过勾选记住我选项登录的只有`IS_AUTHENTICATED_REMEMBERED`这个属性, 但是没有`IS_AUTHENTICATED_FULLY`这个属性。通过账号密码登录的有`IS_AUTHENTICATED_FULLY`这个属性。

然后是`IS_AUTHENTICATED_ANONYMOUSLY`, 所有的用户都有这个属性, 包括未登录的用户都有这个属性。

在5.1版本开始新增了三个属性, `IS_ANONYMOUS`只有匿名的用户有这个属性。`IS_REMEMBERED`只有勾选记住我选项自动登录的用户拥有这个属性。我们看最后一个, 如果当前的用户是模拟的另一个用户, 那么就会有这个属性。复制`IS_AUTHENTICATED_FULLY`这个属性。

回到项目, 我们修改一下`nav.html.twig`文件。复制一行添加注释, 这里修改一下。我们使用`is_granted()`, 参数这里传入`IS_AUTHENTICATED_FULLY`。

回到浏览器刷新首页, 我们再次进行登录。我们再次回到首页, 现在我们的用户信息就显示了出来, 和之前显示的效果一样。我们在开发关于权限的功能时, 通常需要切换账号测试不同的功能, 你需要来回的退出一个账号, 然后登录另一个账号。Symfony为我们提供了另外一种更快速的方法。

回到项目, 我们打开`security.yaml`配置文件。在`main`防火墙下, 有个`switch_user`配置, 我们取消行前的注释。将`switch_user`设置为`true`。我们先来查看一下`switch_user`的文档, 复制链接, 粘贴访问。

通过`switch_user`配置, 我们可以模拟另外一个用户登录, 我们需要在访问的地址后面添加`_switch_user`请求参数, 参数的值就是另外一个用户的标识符。通常使用配置文件中`provider`配置的属性, 这里使用的是用户名。

继续往下看, 我们需要为使用这个功能的用户添加`ROLE_ALLOWED_TO_SWITCH`这个角色属性。通常是用角色继承是个好的办法, 角色继承我们在后面课程会讲到。在这里我们为管理员用户添加这个角色属性, 复制属性的名称。回到项目, 我们打开`Fixtures`文件, 我们为`admin`用户添加这个属性。

重置数据库, 重置完数据库, 回到浏览器, 刷新博客首页。现在我们登录一下超级管理员用户, 点击登录。再次返回首页, 我们的管理员用户已经登录了。我们在链接后面添加下`_switch_user=tom`来模拟tom用户。现在我们看tom用户已经登录了。通过这种方法我们可以快速的切换账户。

继续查看文档, 我们往下看, tom用户现在是模拟的用户, 那么tom用户就会有这个属性(不会读这个单词)。我们可以通过这个属性的判断添加一个退出模拟用户的链接。我们复制一下代码, 回到项目, 打开`nav.html.twig`文件。在菜单后面我们粘贴代码, 首页地址我们修改一下, 修改为`index`。

回到浏览器刷新首页, 现在tom用户后面就添加了退出模拟用户的链接。我们点击链接, 就自动返回了admin用户。回到项目, 我们打开管理端`controller`, 打开`DashboardController`, `DashboardController`继承自`AbstractDashboardController`。

我们查看父类，父类它继承于`AbstractController`，我们再查看`AbstractController`。查看一下类中所有的方法，在`AbstractController`中有一个`getUser()`方法，我们可以通过`getUser()`方法来获取当前已登录的用户。在`getUser()`方法中首先会通过容器获取`Token`服务类对象，再通过`$token`获取用户对象进行返回。

我们还可以通过使用依赖注入的方式，在服务类中或者`controller`方法中获取当前的用户。我们打开`PostController`，在`index()`方法中，我们先来使用父类提供的方法来获取当前的用户，输入`$user = $this->getUser()`。使用`dd`查看`$user`。回到浏览器刷新首页，我们当前的用户就显示了出来。

再次回到项目，我们通过依赖注入的方式注入`Security`对象，这里选择安全系统下的`Security`类型。通过`Security`对象的`getUser()`方法，我们来获取`$user`对象。回到浏览器再次刷新，同样的我们获取到了用户对象。

我们可以使用这两种方法，在`controller`方法中获取当前的用户对象。如果要在服务类中获取用户对象，我们可以在服务类的构造方法中注入`Security`对象，通过`Security`对象来获取用户。

在下节课，我们来增加一些用户的角色，并且学习一下角色的继承。