我们来修改security.yaml配置文件中的access_control配置项，我们取消admin路径这行前的注释。角色这里我们进行一下修改，修改为ROLE_EDITOR和ROLE_CHECKER。然后我们打开Controller类，我们打开DashboardController，注释第25行代码。我们使用配置文件把管理端保护起来。

回到浏览器访问管理端，我们当前的用户是超级管理员，我们来快速的切换一下用户。查看一下editor用户和checker用户是否可以登录管理端，现在切换为editor用户了。它可以访问管理端了，那同样的checker用户也可以访问管理端。

查看一下边栏菜单，我们希望编辑角色的用户和审核角色的用户只能查看文章列表，不能查看评论列表。这里就需要修改DashboardController类的代码。

回到项目，我们需要为边栏菜单设置权限，MenuItem类提供了一个方法，可以为菜单项设置权限。我们来修改评论菜单，使用setPermission()方法为评论菜单添加一个权限，这里我们设置为ROLE_ADMIN。我们只允许管理员及以上级别的用户可以查看评论列表。

```
#src/Controller/Admin/DashboardController.php

class DashboardController extends AbstractDashboardController
{
    /**
     * @Route("/admin", name="admin")
     */
    public function index(): Response
    {
//        if (!$this->isGranted('ROLE_SUPER_ADMIN')){
//            throw new AccessDeniedException();
//        }
//        $this->denyAccessUnlessGranted('ROLE_SUPER_ADMIN');
        return parent::index();
    }

    // ...

    public function configureMenuItems(): iterable
    {
        yield MenuItem::linktoDashboard('Dashboard', 'fa fa-home');
        yield MenuItem::linkToCrud('Post', 'fas fa-list', Post::class);
        yield MenuItem::linkToCrud('Comment', 'fas fa-list',
Comment::class)
        ->setPermission('ROLE_ADMIN');
    }
}
```

回到浏览器刷新，现在评论列表就不显示了，我们再次切换用户。输入simple_admin，现在评论菜单就又显示了。点击文章菜单，我们使用Fixtures类创建了很多文章，但是我们的文章并没有设置作者属性，我们来修改Post类，添加作者属性。

回到项目，输入symfony console make:entity，类名Post，新的属性名称，我们叫做author。文章和作者之间存在着关联关系，一篇文章只有一个作者，但是一个作者可以有多篇文章，文章和作者之间存在着

多对一关系，这里我们选择`ManyToOne`，对应的类我们选择User类，文章的作者可以为空吗？这里不为空。我们需要在User类中添加一个属性，来获取当前用户的所有文章。这里选择yes，User类中新的字段名称叫做posts。直接回车，这里我们需要删除孤儿类型的文章对象吗？选择yes，继续回车退出命令行。

我们查看一下Post类的变更，在Post类中添加了$author属性，然后添加了对应的get和set方法。再查看User类，在User类中添加了$posts属性。同时添加了对应的方法，我们创建数据库的更改文件，执行数据库的更改，这里出错了。我们已经在数据库中创建了一些文章，我们需要为文章添加作者列，但是作者列不允许为空，就触发了外键约束的错误。

我们来清除一下数据库，打开数据库客户端，我们先来删除评论信息，删除所有的评论数据，然后删除所有的文章数据。回到项目，我们再次执行数据库的更改，现在数据库的更改就成功了。

我们来修改一下PostFixtures，为文章添加一些作者，在load()方法中，首先我们通过$manager对象来获取对应角色的用户。获取User类的Repository对象，通过Repository对象来获取对应的用户username。复制29行代码，获取$simpleAdmin用户，审核角色的用户不允许发表文章，我们将这三个用户对象添加到数组中。

在for循环中创建文章时，我们随机的指定作者对象，我们从数组中随机的获取一个索引，然后为$post对象设置作者。我们重新生成一下数据库中的数据，`symfony console doctrine:fixtures:load`，输入yes。

回到浏览器，我们刷新文章列表。这里我们登录一下，现在就重新生成了文章数据，我们需要在列表中显示文章的作者。回到项目，打开PostCrudController类，在`configureFields()`方法中，我们来添加作者列。

文章和作者之间存在着关联关系，我们在body后面添加Association::new()。属性名称输入author。回到列表刷新，现在作者列就显示了，但是显示的是作者的id。

回到项目，我们修改User类实现一下`__toString()`方法，返回作者的用户名。再次回到列表刷新，现在作者列就显示了，我们再来切换一下用户，在地址栏后面我们添加`_switch_user=editor`。

现在就是editor用户，我们来编辑其他用户的文章，它可以编辑其他用户的文章。返回，我们希望文章的作者只能编辑或者删除自己编写的文章。

在下节课我们将学习自定义Voter，使用Voter来完成这项功能。