

我们查看一下Symfony Security文档，看一下文档的目录。这里有一段是关于层级角色的，我们点击这里。在security配置文件中，我们可以配置角色的层级，这里有段示例的配置。我们看一下配置的文档。

ROLE_ADMIN角色会拥有ROLE_USER角色的权限。

ROLE_SUPER_ADMIN角色会自动的拥有ROLE_ADMIN角色、ROLE_ALLOWED_TO_SWITCH角色和ROLE_USER角色。

ROLE_SUPER_ADMIN角色从ROLE_ADMIN角色进行了继承。

我们复制配置项，回到项目，打开security.yaml配置文件。我们添加一下角色的继承，随着我们博客系统的开发，我们可能需要为用户配置多个角色。首先我们配置一个编辑的角色ROLE_EDITOR，它继承于ROLE_USER。然后是文章审核角色ROLE_CHECKER，它也继承于ROLE_USER角色。然后是普通管理员角色，普通管理员角色继承于上面两个角色，拥有这两个角色的权限，使用中括号，继承于ROLE_EDITOR和ROLE_CHECKER。最后是超级管理员角色，超级管理员角色继承于ROLE_ADMIN角色，ROLE_SUPER_ADMIN继承于ROLE_ADMIN，然后超级管理员角色还可以使用switch user，这里我们需要添加一下ROLE_ALLOWED_TO_SWITCH。

```
#config/packages/security.yaml
security:
    role_hierarchy:
        ROLE_EDITOR: ROLE_USER
        ROLE_CHECKER: ROLE_USER
        ROLE_ADMIN: [ROLE_EDITOR, ROLE_CHECKER]
        ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]

# ...
```

这样我们的角色层级就配置好了，我们来修改Fixtures文件。来创建几个用户，并且指定用户的角色。打开Fixtures，我们先来修改一下admin用户。现在admin用户只需要指定ROLE_SUPER_ADMIN角色就可以了。

我们复制一下这里的代码，粘贴。我们创建一个编辑用户editor，设置编辑用户的用户名editor，设置它的角色为ROLE_EDITOR，设置密码，密码我们也设置为editor。

然后我们添加审核用户，checker，用户名我们修改为checker，角色修改为ROLE_CHECKER，密码也修改为checker (这里参数\$editor应该修改为\$checker)。

然后我们添加一个普通管理员用户\$simpleAdmin，用户名我们设置为simple_admin，角色设置为ROLE_ADMIN，然后密码也设置为admin。最后我们使用\$manager对象进行一下保存，重置一下数据库。

```
#src/DataFixtures/AppFixtures.php
class AppFixtures extends Fixture
{
    public function load(ObjectManager $manager)
    {
        // ...

        $editor = new User();
        $editor->setUsername('editor');
        $editor->setRoles(['ROLE_EDITOR']);
        $editor->setPassword($this->userPasswordHasher->
```

```
>hashPassword($editor, 'editor'));

    $checker = new User();
    $checker->setUsername('checker');
    $checker->setRoles(['ROLE_CHECKER']);
    $checker->setPassword($this->userPasswordHasher-
>hashPassword($checker, 'checker'));

    $simpleAdmin = new User();
    $simpleAdmin->setUsername('simple_admin');
    $simpleAdmin->setRoles(['ROLE_ADMIN']);
    $simpleAdmin->setPassword($this->userPasswordHasher-
>hashPassword($simpleAdmin, 'admin'));

    // ...

    $manager->persist($editor);
    $manager->persist($checker);
    $manager->persist($simpleAdmin);

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

打开模板文件, 我们修改一下base.html.twig文件。在content区块上方, 我们添加一些代码。`if is_granted('ROLE_EDITOR'), endif`。这里我们显示一下当前用户的角色, 我们快速的复制, 并且编辑一下代码。

```
#templates/base.html.twig
<!-- Blog Entries Column -->
<div class="col-md-8">
    {% if is_granted('ROLE_EDITOR') %}
        当前用户是编辑角色
    {% endif %}
    {% if is_granted('ROLE_CHECKER') %}
        当前用户是审核角色
    {% endif %}
    {% if is_granted('ROLE_ADMIN') %}
        当前用户是管理员角色
    {% endif %}

    {% block content %}
    {% endblock %}
</div>
```

回到浏览器, 我们刷新一下首页。现在我们登录一下用户查看一下首页的显示。我们首先登录超级管理员, 回到首页, 超级管理员拥有所有的角色。所以它既是编辑角色, 又是审核角色, 又是管理员角色。

我们进行一下用户的切换输入`_switch_user`, 先切换为编辑用户editor, 它只显示编辑角色。我们再次进行切换, 这次我们切换为审核用户, 显示为当前为审核角色。我们再次进行切换, 这次我们切换为普通管理

员用户，普通管理员角色，它继承了编辑角色和审核角色，所以它拥有三个角色。

这就是角色的继承，我们退出模拟用户，回到超级管理员用户。我们的编辑角色、审核角色、管理员角色他们应该都能访问管理端。在下节课我们来修改管理端的代码，让这些用户可以访问管理端，并且添加一些权限。