

Symfony提供了安全系统来对用户的登录进行认证,我们并不需要手动的创建表单,并且在controller方法中获取用户名密码来进行验证。我们查看一下所有的make命令行,这里有个`make:auth`命令。这里提示我们可以创建一个受保护的认证器,我使用这个命令行。

命令行提示我们要创建哪种类型的认证器,第一种是个空的认证器,第二种是个登录表单的认证器。如果我们在创建一个API系统,我们就需要选择一个空的认证器。但是现在我们需要有个登录表单,我们就选择下面这一个,输入`1`。

这里提示我们创建一个认证器的类名,我们叫做`FormLoginAuthenticicator`,这里让我们选择一个Controller名称,我们选择默认的就就可以了。回车,它其实我们需要生成一个`/logout`路径吗,我们选择yes。命令行为我们创建了三个文件,分别是认证器类,Controller类和登录界面模板,然后更新了security.yaml配置文件。

我们来查看一下这几个文件,先来查看Controller文件,在Controller类中分别为我生成了两个controller方法。第一个是用于用户登录的方法。第二个是用于用户退出的方法。

我们再来查看一下配置文件,在配置文件中,在防火墙main配置键下,添加了一个`custom_authenticator`配置,将我们添加的认证器类进行了配置,并且添加了一个`logout`设置,退出的路由设置为`app_logout`路由。

```
#config/packages/security.yaml

security:
    # ...
    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false
        main:
            lazy: true
            provider: app_user_provider
            custom_authenticator: App\Security\FormLoginAuthenticicator
            logout:
                path: app_logout
                # where to redirect after logout
                # target: app_any_route
```

我们再来查看一下认证器类,在Security文件夹下打开这个类,在这个类中有一个`authenticate()`方法,它会获取请求中的用户名,然后返回了一个`Passport`对象。

```
#src/Security/FormLoginAuthenticator.php

class FormLoginAuthenticator extends AbstractLoginFormAuthenticator
{
    use TargetPathTrait;

    public const LOGIN_ROUTE = 'app_login';
```

```
private UrlGeneratorInterface $urlGenerator;

public function __construct(UrlGeneratorInterface $urlGenerator)
{
    $this->urlGenerator = $urlGenerator;
}

public function authenticate(Request $request): PassportInterface
{
    $username = $request->request->get('username', '');

    $request->getSession()->set(Security::LAST_USERNAME, $username);

    return new Passport(
        new UserBadge($username),
        new PasswordCredentials($request->request->get('password',
        '')),
        [
            new CsrfTokenBadge('authenticate', $request-
            >get('_csrf_token')),
        ]
    );
}

public function onAuthenticationSuccess(Request $request,
TokenInterface $token, string $firewallName): ?Response
{
    if ($targetPath = $this->getTargetPath($request->getSession(),
$firewallName)) {
        return new RedirectResponse($targetPath);
    }

    // For example:
    return new RedirectResponse($this->urlGenerator-
>generate('admin'));
    // throw new \Exception('TODO: provide a valid redirect inside
    '.__FILE__');
}

protected function getLoginUrl(Request $request): string
{
    return $this->urlGenerator->generate(self::LOGIN_ROUTE);
}
```

再往下有个on方法，见名知意，当表单通过验证时会调用这个方法。如果通过的话，会获取一个\$targetPath。如果\$targetPath存在的话，就会跳转到\$targetPath。否则的话我们可以跳转到一个自定义路由。

这里我们修改一下54行代码，取消注释，注释55行代码，我们让它跳转到管理端的首页admin路由。最后是getLoginUrl()方法，这个方法用于返回登录页面的路径。

打开templates目录, 在security目录中我们打开login.html.twig文件, 它继承于base.html.twig模板。我们需要修改body区块名称, 这里我们输入content。

```
#templates/security/login.html.twig

{% extends 'base.html.twig' %}

{% block title %}Log in!{% endblock %}

{% block content %}
<form method="post">
    {% if error %}
        <div class="alert alert-danger">{{
error.messageKey|trans(error.messageData, 'security') }}</div>
    {% endif %}

    {% if app.user %}
        <div class="mb-3">
            You are logged in as {{ app.user.username }}, <a href="{{
path('app_logout') }}">Logout</a>
        </div>
    {% endif %}

    <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
    <label for="inputUsername">Username</label>
    <input type="text" value="{{ last_username }}" name="username"
id="inputUsername" class="form-control" autocomplete="username" required
autofocus>
    <label for="inputPassword">Password</label>
    <input type="password" name="password" id="inputPassword" class="form-
control" autocomplete="current-password" required>

    <input type="hidden" name="_csrf_token"
        value="{{ csrf_token('authenticate') }}"
    >

    {#
        Uncomment this section and add a remember_me option below your
        firewall to activate remember me functionality.
        See https://symfony.com/doc/current/security/remember_me.html
    <div class="checkbox mb-3">
        <label>
            <input type="checkbox" name="_remember_me"> Remember me
        </label>
    </div>
    #}

    <button class="btn btn-lg btn-primary" type="submit">
        Sign in
    </button>
</form>
{% endblock %}
```

回到浏览器，我们来访问一下登录页面，输入`127.0.0.1:8000/login`，在login页面中就有了登录的表单框。我们输入`admin/admin`，点击登录。点击登录之后，它自动跳转到管理端的首页。

我们看底部的工具栏，这里我们登录的用户就是admin。我们一行代码都没有编写，但是我们已经实现了用户登录的所有功能。

我们退出当前的用户，再次访问管理端，依然可以访问，我们的管理端并没有保护起来。回到项目我们修改一下`security.yaml`配置文件，在`access_control`配置键下，我们为管理端的路径添加一个权限，角色修改为`ROLE_SUPER_ADMIN`，只有`ROLE_SUPER_ADMIN`超级管理员才可以登录。

回到管理端，我们再次刷新管理端的首页。我们当前没有登录用户，Symfony自动的将我们的页面跳转到了登录页面。这就是认证器类`getLoginUrl()`方法的作用，我们可以在这个方法中来自定义登录的链接。

在下节课。我们来详细的了解一下用户登录的流程。