

我们搜索`symfony remember me`, 查看一下symfony文档。我们打开第一个页面, 在文档中有段配置文件的代码, 在main防火墙下添加了`remember_me`设置。通过这个设置来增加Remember me功能。我们继续往下看, `remember_me`有很多选项可以进行设置。这里我们复制示例的配置。回到项目, 打开`security.yaml`配置文件, 在main防火墙下我们添加配置, 粘贴。

```
#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\FormLoginAuthenticator
            remember_me:
                secret: '%kernel.secret%'
                lifetime: 604800 # 1 week in seconds
                path: /

    # ...
```

我们看`secret`配置, 当我们登录之后, Symfony会将我们的账户信息进行一些加密, 然后保存到cookie中。当我们下次登录时浏览器会调用我们的cookie进行登录。cookie中的数据需要加密, 它使用了Symfony内置的参数`kernel.secret`进行加密。cookie保存的时间, 这里设置的是一星期。当访问项目的所有路径时都会使用Remember me功能进行自动登录。

回到浏览器, 打开login页面, 我们登录一下, 勾选Remember me登录, 现在我们就登录成功了。现在我们关闭浏览器, 退出浏览器, 重新打开浏览器, 再次访问博客系统。我们看到我们的用户并没有自动登录, 我们使用了新的认证系统, 我们还需要在认证系统中进行一些配置。

回到项目, 打开我们自己的Authenticator类, 在`authenticate()`方法中, 最后需要返回一个`Passport`对象。`Passport`对象有三个参数, 第三个参数是徽章的数组, 我们需要传递一个徽章。回到Authenticator类, 我们需要增加一个Remember me徽章。

```
class FormLoginAuthenticator extends AbstractLoginFormAuthenticator
{
    // ...

    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')),  
            new RememberMeBadge()  
        ]  
    );  
}  
  
// ...  
}
```

回到浏览器，我们再次登录。勾选Remember me，登录。我们退出浏览器，再次打开浏览器，再次访问博客首页。现在的用户就可以自动登录了，这就是新版认证系统，Remember me的设置。

在下节课，我们来学习另一个有趣的功能。