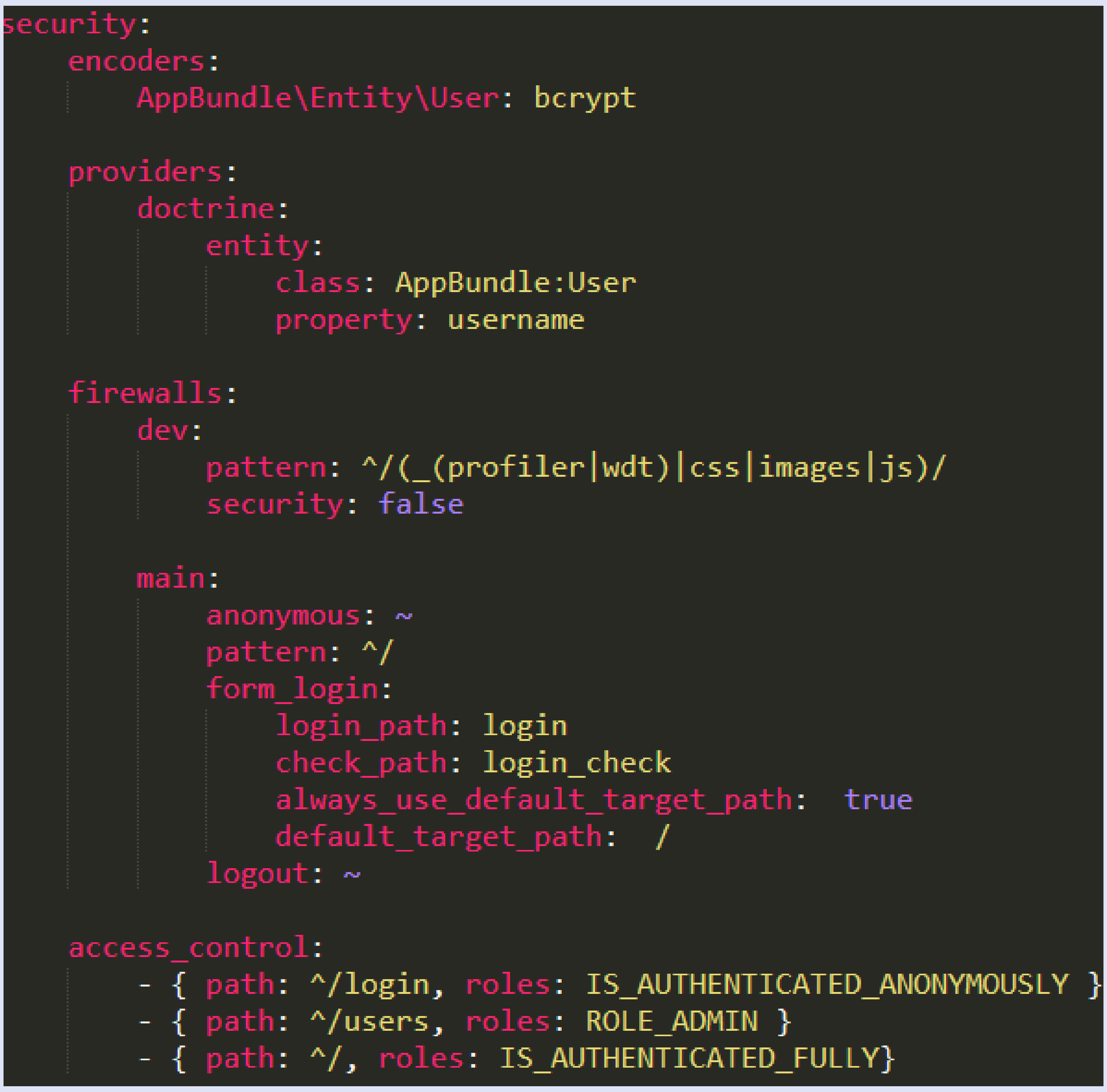
TECHNICAL DOCUMENTATION

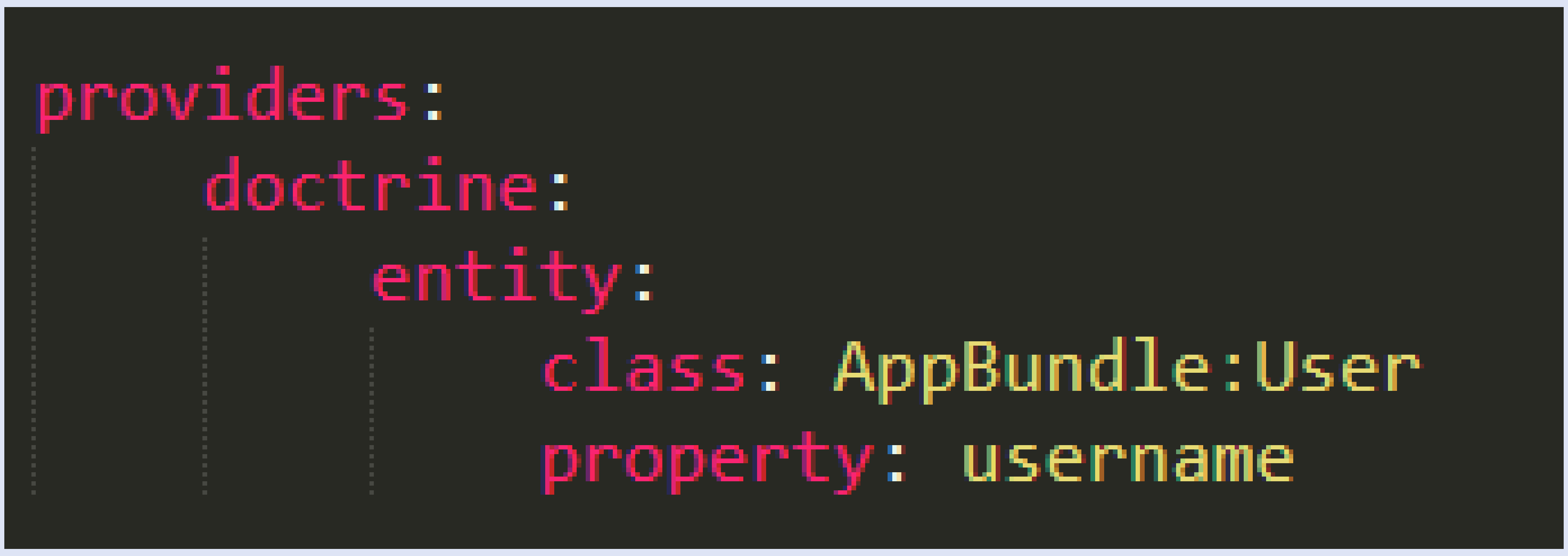
NEW USERS

1) THE SECURITY FILES IS LOCATED IN APP/CONFIG/SECURITY.YML, THIS IS THE FILE THAT NEED TO BE MODIFIED WHEN WE WANT TO ADD RESTRICTIONS TO ACCESS THE APP.



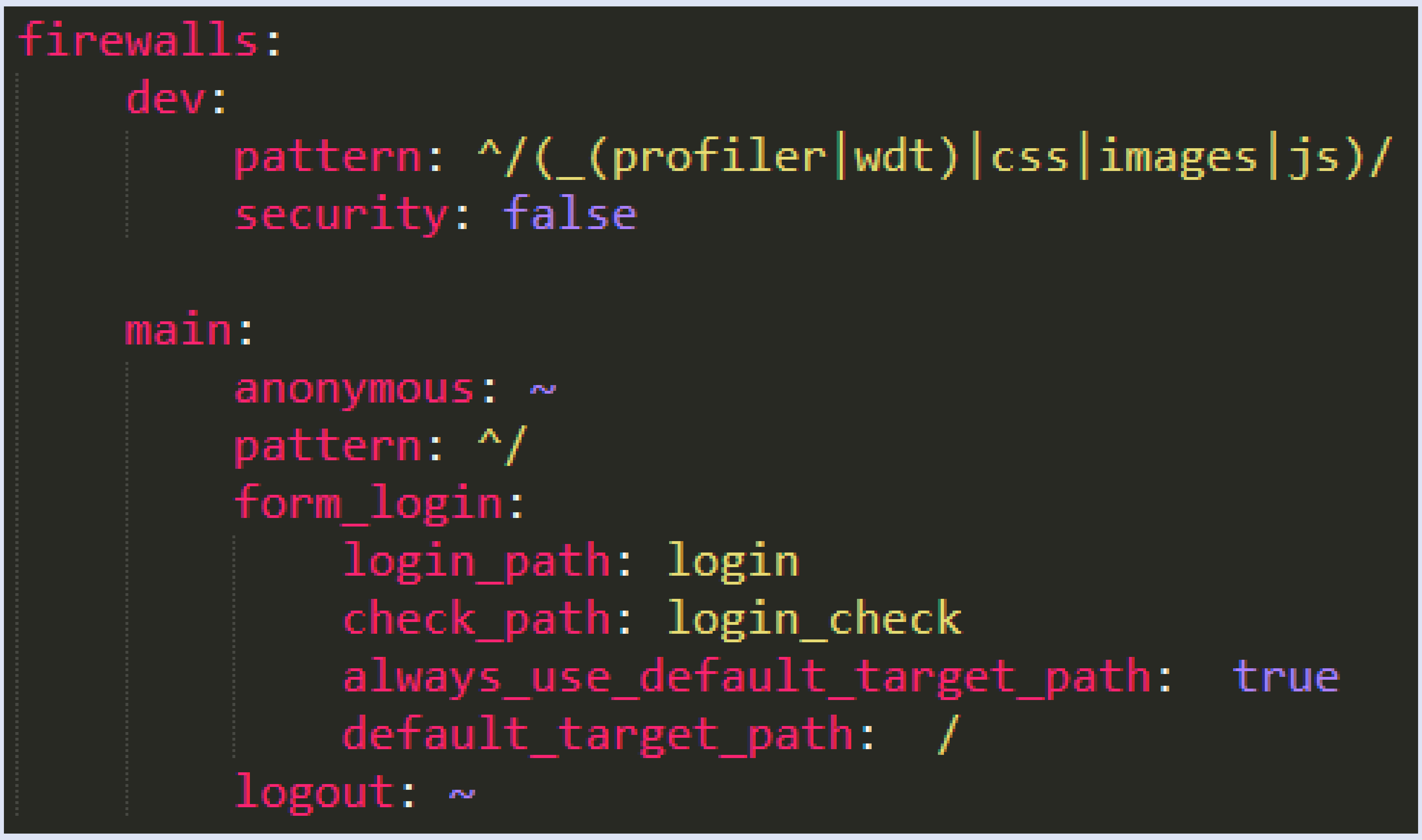
```
security:
    encoders:
        AppBundle\Entity\User: bcrypt

    providers:
        doctrine:
            entity:
                class: AppBundle:User
                property: username

    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false

        main:
            anonymous: ~
            pattern: ^/
            form_login:
                login_path: login
                check_path: login_check
                always_use_default_target_path:  true
                default_target_path:  /
            logout: ~

    access_control:
        - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/users, roles: ROLE_ADMIN }
        - { path: ^/, roles: IS_AUTHENTICATED_FULLY}
```

2) THE USERS ARE LOADED FROM THE DATABASE
(AS DISCRIBED IN THIS PICTURE)

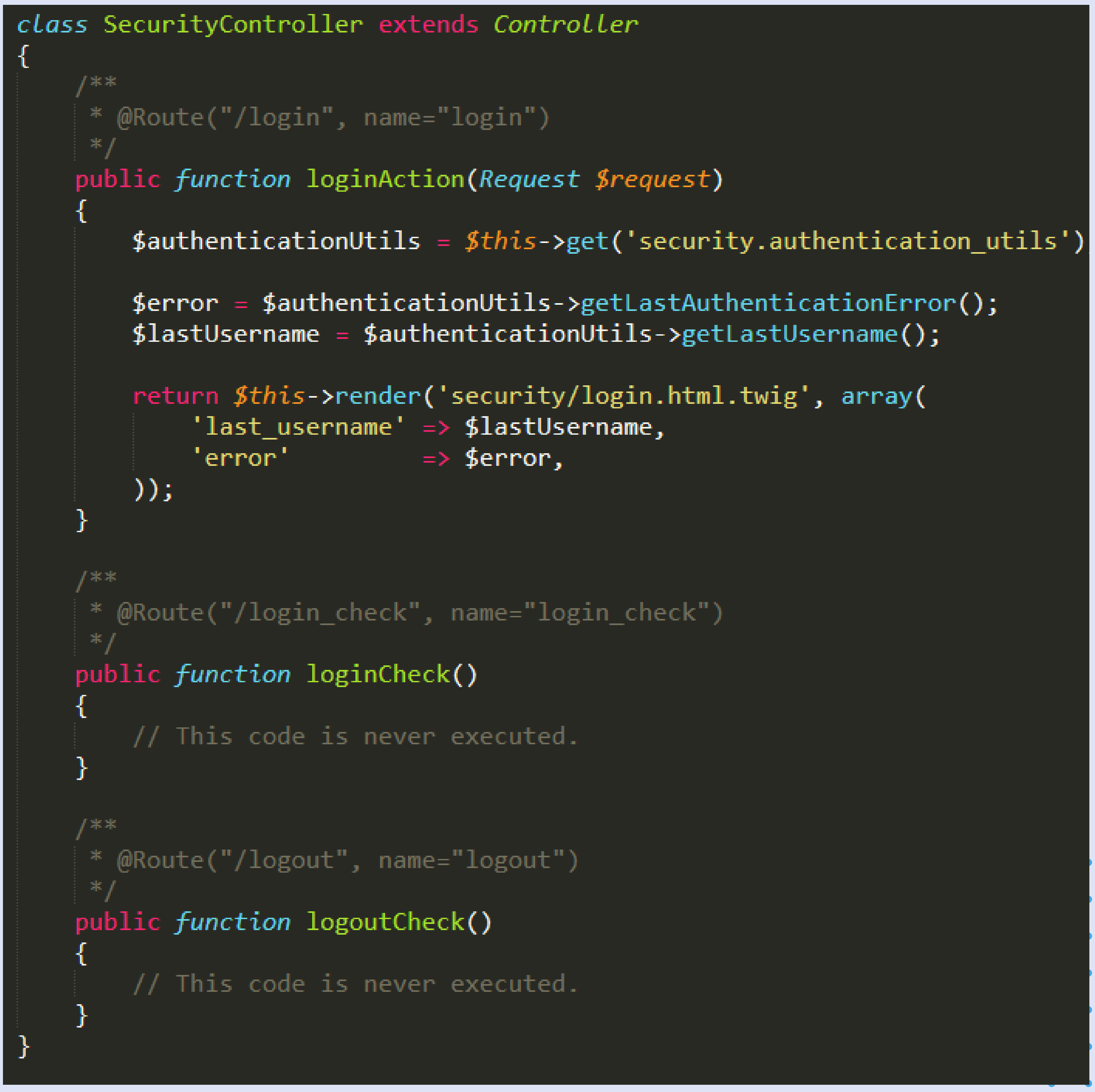


```
providers:
    doctrine:
        entity:
            class: AppBundle:User
            property: username
```

3) all the users are by default redirected to the login form page (see img 3) and the authentification are managed by the SecurityController.php file (see img 4).

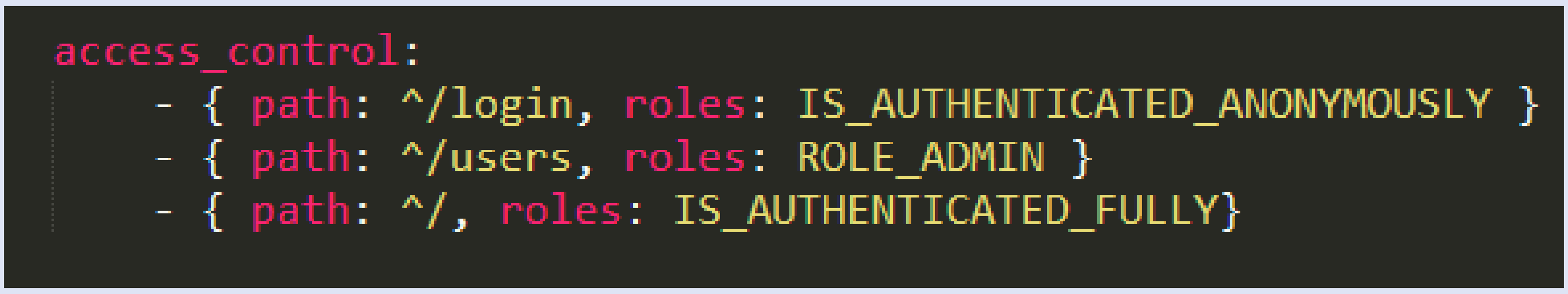


```yaml
firewalls:
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false

    main:
        anonymous: ~
        pattern: ^/
        form_login:
            login_path: login
            check_path: login_check
            always_use_default_target_path:  true
            default_target_path:  /
        logout: ~
```

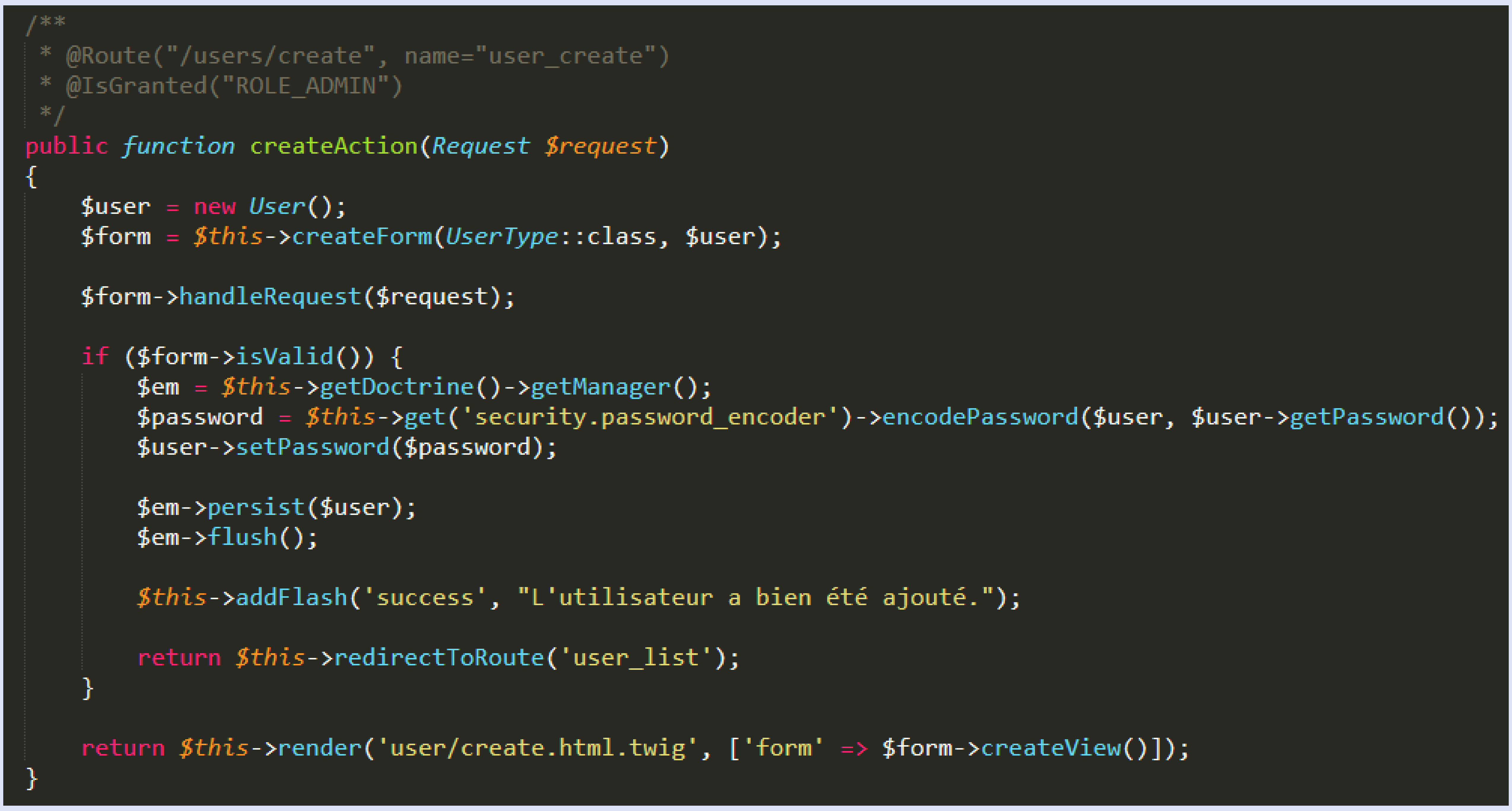


```php
class SecurityController extends Controller
{
    /**
     * @Route("/login", name="login")
     */
    public function loginAction(Request $request)
    {
        $authenticationUtils = $this->get('security.authentication_utils')

        $error = $authenticationUtils->getLastAuthenticationError();
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('security/login.html.twig', array(
            'last_username' => $lastUsername,
            'error'         => $error,
        ));
    }

    /**
     * @Route("/login_check", name="login_check")
     */
    public function loginCheck()
    {
        // This code is never executed.
    }

    /**
     * @Route("/logout", name="logout")
     */
    public function logoutCheck()
    {
        // This code is never executed.
    }
}
```

4)all the users can acess the login page but only authentificated users can use and access the app functionalities and finaly only admins can access the user page management as seen here.



```
access_control:
    - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/users, roles: ROLE_ADMIN }
    - { path: ^/, roles: IS_AUTHENTICATED_FULLY}
```



```
/**
 * @Route("/users/create", name="user_create")
 * @IsGranted("ROLE_ADMIN")
 */
public function createAction(Request $request)
{
    $user = new User();
    $form = $this->createForm(UserType::class, $user);

    $form->handleRequest($request);

    if ($form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $password = $this->get('security.password_encoder')->encodePassword($user, $user->getPassword());
        $user->setPassword($password);

        $em->persist($user);
        $em->flush();

        $this->addFlash('success', "L'utilisateur a bien été ajouté.");

        return $this->redirectToRoute('user_list');
    }

    return $this->render('user/create.html.twig', ['form' => $form->createView()]);
}
```