

DASHBOARDCONTROLLER.PHP CODE

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class DashboardController extends AbstractController
{
    #[Route('/dashboard', name: 'app_dashboard')]
    public function index(): Response
    {
        return new Response("Welcome to the dashboard! (Protected Area)");
    }
}
```

SECURITY.YAML CODE

```
security:
#
https://symfony.com/doc/current/security.html#registering-the-user-hashing
-passwords
    password_hashers:

Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
    'auto'
#
https://symfony.com/doc/current/security.html#loading-the-user-the-user-pr
ovider
    providers:
        users_in_memory: { memory: null }
    firewalls:
```

```

dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
main:
    lazy: true
    provider: users_in_memory

    # activate different ways to authenticate
    # https://symfony.com/doc/current/security.html#the-firewall

#
https://symfony.com/doc/current/security/impersonating_user.html
    # switch_user: true

# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
    # - { path: ^/admin, roles: ROLE_ADMIN }
    # - { path: ^/profile, roles: ROLE_USER }
    - { path: ^/dashboard, roles: ROLE_USER }

when@test:
    security:
        password_hashers:
            # By default, password hashers are resource intensive and take
time. This is
            # important to generate secure password hashes. In tests
however, secure hashes
            # are not important, waste resources and increase test times.
The following
            # reduces the work factor to the lowest possible values.

Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
    algorithm: auto
    cost: 4 # Lowest possible value for bcrypt
    time_cost: 3 # Lowest possible value for argon
    memory_cost: 10 # Lowest possible value for argon

```