A Basic Controller

While a controller can be any PHP callable (function, method on an object, or a Closure), a controller is usually a method inside a controller class:

*// src/Controller/LuckyController.php*

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

class LuckyController

{

*/\*\**

*\* @Route("/lucky/number/{max}", name="app\_lucky\_number")*

*\*/*

public function number(int $max): Response

{

$number = random\_int(0, $max);

return new Response(

'<html><body>Lucky number: '.$number.'</body></html>'

);

}

}

Le contrôleur est la méthode du nombre() qui vit à l’intérieur de la classe de contrôleur LuckyController.

Ce contrôleur est assez simple:

ligne 2 : Symfony tire parti des fonctionnalités d’espace nominatif de PHP pour nommer l’ensemble de la classe de contrôleur.

ligne 4 : Symfony profite à nouveau de la fonctionnalité namespace de PHP : le mot clé d’utilisation importe la classe Réponse, que le contrôleur doit retourner.

ligne 7: La classe peut techniquement être appelé n’importe quoi, mais il est suffixe avec contrôleur par convention.

ligne 12: La méthode d’action est autorisée à avoir un argument $max grâce à la wildcard {max} dans la route.

ligne 16 : Le contrôleur crée et renvoie un objet Réponse.

Cartographier une URL à un contrôleur¶

Afin d’afficher le résultat de ce contrôleur, vous devez y cartographier une URL via un itinéraire. Cela a été fait ci-dessus avec @Route (« /lucky/number/{max} ») annotation de l’itinéraire.

Pour voir votre page, rendez-vous sur cette URL dans votre navigateur : http://localhost:8000/lucky/number/100

Pour plus d’informations sur le routage, voir Routing.

La classe et services du contrôleur de base

Pour faciliter le développement, Symfony est livré avec une classe de contrôleur de base optionnelle appelée Symfony\Bundle\FrameworkBundle\Controller\AbstractController. Il peut être étendu pour accéder aux méthodes d’aide.

Ajoutez l’instruction d’utilisation au sommet de votre classe de contrôleur, puis modifiez LuckyController pour l’étendre :

// src/Controller/LuckyController.php

namespace App\Controller;

+ use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

- class LuckyController

+ class LuckyController extends AbstractController

{

// ...

}

That’s it! You now have access to methods like $this->render() and many others that you’ll learn about next.

Generating URLs

La méthode generateUrl () n’est qu’une méthode d’aide qui génère l’URL d’un itinéraire donné :

$url = $this->generateUrl('app\_lucky\_number', ['max' => 10]);

Redirection

Si vous souhaitez rediriger l’utilisateur vers une autre page, utilisez les méthodes redirecttoroute () et redirection () :

use Symfony\Component\HttpFoundation\RedirectResponse;

*// ...*

public function index(): RedirectResponse

{

*// redirects to the "homepage" route*

return $this->redirectToRoute('homepage');

*// redirectToRoute is a shortcut for:*

*// return new RedirectResponse($this->generateUrl('homepage'));*

*// does a permanent - 301 redirect*

return $this->redirectToRoute('homepage', [], 301);

*// redirect to a route with parameters*

return $this->redirectToRoute('app\_lucky\_number', ['max' => 10]);

*// redirects to a route and maintains the original query string parameters*

return $this->redirectToRoute('blog\_show', $request->query->all());

*// redirects externally*

return $this->redirect('http://symfony.com/doc');

}

La méthode de redirection () ne vérifie en aucune façon sa destination. Si vous redirigez vers une URL fournie par les utilisateurs finaux, votre application peut être ouverte à la vulnérabilité de sécurité redirige non garantie.

Modèles de rendu¶

Si vous servez HTML, vous voudrez rendre un modèle. La méthode de rendu () rend un modèle et met ce contenu dans un objet réponse pour vous :

return $this->render('lucky/number.html.twig', ['number' => $number]);