

Symfony

LUCKY NUMBER Y COIN

Roberto Martínez Avendaño

Introducción a Symfony

Gracias a Symfony se ahorra mucho código utilizando el Response para enviar respuestas HTTP de tipo POST y Route para crear el enrutado de la aplicación.

Lucky Number

Código

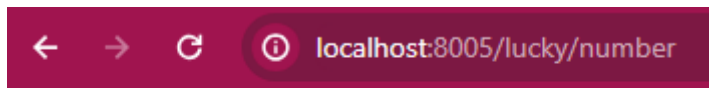
```
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

0 references | 0 implementations
class LuckyController
{
    /*El componente route forma la url del programa,
    por lo que acceder a la función se vuelve más simple*/
    #[Route('/lucky/number')]
    1 reference | 0 overrides
    public function number(): Response //Response es un objeto el cual se envia al navegador
    {
        $number = random_int(0, 100);

        return new Response(
            '<html><body>Lucky number: '.$number.'</body></html>'
        );
    }
}
```

En el **LuckyController** imprimimos por pantalla un numero aleatorio entre 0 y 100.

Ejecución



Lucky number: 40

Coin

Código

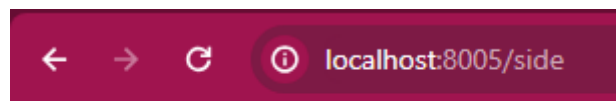
```
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

0 references | 0 implementations
class coinController
{
    /*El componente route forma la url del programa,
    por lo que acceder a la funcion se vuelve más simple*/
    #[Route('/side')]
    1 reference | 0 overrides
    public function flipCoin(): Response //Response es un objeto el cual se envia al navegador
    {
        $cara = random_int(0, 1);

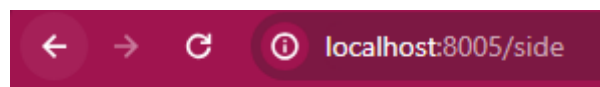
        return new Response(
            '<html><body>Ha salido: ' . (($cara) ? 'cara' : 'cruz') . '</body></html>'
        );
    }
}
```

En el **coinController** utilizamos una ternaria para definir si el resultado que se va a imprimir por pantalla será cara o cruz en función de si la variable cara genera un resultado true o false (1 o 0).

Ejecución



Ha salido: cruz



Ha salido: cara