

Symfony 3

jesusdasilva

Published
with GitBook



Table of Contents

Introduction	0
First Chapter	1
Controladores	2

My Awesome Book

This file serves as your book's preface, a great place to describe your book's content and ideas.

First Chapter

GitBook allows you to organize your book into chapters, each chapter is stored in a separate file like this one.

Controladores

Un controlador es una función PHP creada por ti, cuyo propósito es obtener la información de la solicitud HTTP para generar y devolver una respuesta HTTP. La respuesta podría ser una página HTML, un documento XML, un array JSON serializado, una imagen, una redirección a otra página, un error 404 o cualquier otra cosa. El controlador contiene cualquier lógica necesaria para procesar el contenido de una página.

Ahora en acción un simple controlador de Symfony. Esto hace que una página imprima el famoso Hola!:

```
use Symfony\Component\HttpFoundation\Response;

public function helloAction()
{
    return new Response('Hello world!');
}
```

El objetivo de un controlador es siempre el mismo: crear y devolver un objeto Respuesta. En el camino, se podría leer la información de la solicitud, cargar un recurso de base de datos, enviar un correo electrónico, o la información que figura en la sesión del usuario. Pero en todos los casos, el controlador finalmente devolverá el objeto de Respuesta que será entregado de nuevo al cliente.

No es magia y no hay otros requisitos de que preocuparse! Aquí hay algunos ejemplos comunes:

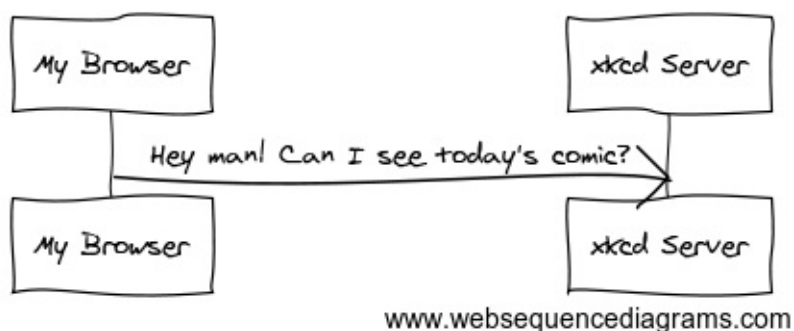
- Un controlador prepara un objeto *Response* que representa el contenido de la página principal del sitio.
- El controlador B lee el *{slug}* de la petición para cargar una entrada del *Blog* desde una base de datos y crea un objeto Respuesta mostrándolo. Si el *{slug}* no se puede encontrar en la base de datos, se devuelve un objeto Respuesta con un código de estado 404.
- El Controlador C maneja el envío de un formulario de contacto. Se lee la información de la solicitud, se guarda la información de contacto en la base de datos y se envía un mensajes de correo electrónico con la información de contacto a usted. Por último, se crea un objeto Respuesta que redirige el navegador del cliente a una "página de agradecimientos".

Solicitud, Controlador y Ciclo de Vida de una Respuesta.

En los proyectos symfonys cada petición pasa por un mismo y simple ciclo de vida. El Framework se encarga de todas las tareas repetitivas y Uds solo escribe su código personalizado en la función del controlador:

1. cada petición ejecuta un solo archivo que es el controlador frontal (`app.php` si esta en producción o `app_dev.php` en desarrollo) al iniciar la aplicación;
2. El único trabajo del controlador frontal es inicializar el motor de Symfony (el Kernel) y pasar un Objeto *Request* para su manejo;
3. El núcleo de Symfony pide la ruta para inspeccionar la petición.
4. La ruta coincide con una URL de una ruta específica y devuelve la información sobre la ruta, incluyendo el controlador que debe ejecutarse.
5. El controlador que concuerde con la ruta buscada es ejecutado y el código dentro del controlador crea y retorna el Objeto *Response* apropiado.
6. Las cabeceras HTTP y el contenido del objeto de *Response* se envían de vuelta al cliente.

La creación de una página es tan fácil como crear un controlador (# 5) y hacer una ruta que asigna una dirección URL a ese controlador (# 4).



Aunque el nombre es similar, un "controlador frontal" es diferente a las funciones de PHP llamadas "controladores" que se hablan en este capítulo. Un controlador frontal es un pequeño archivo PHP que se encuentra en el directorio web/ directorio a través del cual se dirigen todas las peticiones.

Una aplicación típica tendrá un controlador frontal para producción (por ejemplo, `app.php`) y un controlador frontal para desarrollo (por ejemplo `app_dev.php`). Es probable que nunca tenga que volver a editar, ver o preocuparse por los controladores frontales de su aplicación. La "clase controlador" es una manera conveniente de agrupar varios "controladores", también llamadas acciones, en una sola clase (por ejemplo `updateAction()`, `DeleteAction()`, etc.). Por lo tanto, un controlador es un método dentro de una clase controlador. Ellos tienen su código que crea y devuelve el Objeto *Response* adecuado.

Un Simple Controlador

Mientras que un controlador puede ser cualquier elemento PHP ejecutable (una función, un método en un objeto, o una función anónima), un controlador es por lo general un método dentro de una clase controlador:

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

```
// src/AppBundle/Controller/HelloController.php
namespace AppBundle\Controller;

use Symfony\Component\HttpFoundation\Response;

class HelloController
{
    public function indexAction($name)
    {
        return new Response('<html><body>Hello '.$name.'!</body></html>');
    }
}
```