

Configuración

En Symfony4 siguen siendo válidos los 3 formatos de ficheros de configuración: YAML, PHP y XML siendo el preferido y el utilizado por defecto el YAML.

La configuración de cada paquete se encuentra en **config/packages**. Por ejemplo, el FrameworkBundle se configura en **config/packages/framework.yaml**

Los ficheros están organizados de otra forma, pero siguen el mismo sistema que en Symfony3.

Cada bundle/package tiene su clave de primer nivel.

Sigue existiendo además la sección *parameters* para parametrizar valores.

La novedad en Symfony4 son las llamadas variables de entorno.

Para conocer las opciones de configuración de un Bundle tenemos dos opciones:

- Acudir a la documentación oficial de dicho bundle
- Ejecutar el comando `config:dump-reference`

```
php bin/console config:dump-reference framework
```

Entornos y variables de entorno

Las variables de entorno y el fichero .env

En el directorio raíz hay un fichero .env en el que se definen variables dependientes del entorno y/o del servidor: desarrollo, producción, test, etc.

Este fichero NO se sube al control de versiones. Hay un fichero .env.dist con valores dummy, que sirve como referencia al programador para saber que valores deben aparecer en el fichero .env.

Ejemplo de uso de una variable de entorno:

```
# config/packages/doctrine.yaml
doctrine:
    dbal:
        url: '%env(DATABASE_URL)%'
```

Entornos

<http://symfony.com/doc/master/configuration/environments.html>

En el raíz de `config/packages` se define la configuración común para todos los entornos.

En las subcarpetas `dev`, `prod` y `test`, se definen las configuraciones que difieren en cada entorno.

- Para el entorno *dev*: `config/packages/dev/`
- Para el entorno *prod*: `config/packages/prod/`
- Para el entorno *test*: `config/packages/test/`

En el fichero `src/Kernel.php` se puede cambiar la localización de estos ficheros.

Ejecutar Symfony con un entorno específico

Para ejecutar la aplicación en un entorno concreto, basta con cambiar la variable **APP_ENV** del fichero `.env`.

```
# .env  
APP_ENV=test
```

Si inspeccionamos el fichero `public/index.php` veremos cómo funciona la variable:

```
// public/index.php
```

```
// ...  
$kernel = new Kernel($_SERVER['APP_ENV'] ?? 'dev', $_SERVER['APP_DEBUG'] ?? false);  
  
// ...
```

El segundo parámetro de Kernel es si se ejecuta en modo debug.

La misma técnica se utiliza en el script bin/console:

```
$input = new ArgvInput();  
$env = $input->getParameterOption(['--env', '-e'], $_SERVER['APP_ENV'] ?? 'dev', true);  
$debug = (bool) ($_SERVER['APP_DEBUG'] ?? ('prod' !== $env)) && !$input->hasParameterOption('--no-debug', true);  
  
// ...  
  
$kernel = new Kernel($env, $debug);  
$application = new Application($kernel);  
$application->run($input);
```

Creación de un entorno personalizado

Por defecto, Symfony 4 viene con los 3 entornos de sobra conocidos por todos: prod, dev y test. No obstante, al igual que en la versión 3, en la 4 podemos crear nuestro propio entorno personalizado.

Crear un entorno

Para crear un entorno, basta con crear un directorio en *config/packages* con el nombre del entorno y poner allí la configuración específica del entorno.

Podemos reutilizar archivos de configuración utilizando la clave *imports* ya existente en symfony 3.

```
imports:
    - { resource: .../dev/monolog.yaml }
    - { resource: .../dev/swiftmailer.yaml }
```