

Introducció a Symfony

1. Instal·lació y configuració de Symfony
 - i. Arquitectura MVC
 - ii. Introducció a YAML
2. Vistes
 - i. Motor de plantilles Twig
3. Injecció de dependències: Contenedors
4. El model
 - i. Doctrine
5. Creació i validació de formularis
6. Seguretat i control d'accés
7. Symfony *Bundles*

Frameworks PHP

Un **framework** és una eina que proporciona una sèrie de mòduls que ajuden a organitzar i desenvolupar un producte de programari.

La majoria dels *frameworks* PHP proporcionen una sèrie de comandos o eines per a crear projectes amb una estructura determinada (normalment seguint el patró MVC), de manera que ja donen una base de treball feta, i facilitats per a poder crear el model de dades, la connexió a la base de dades, les rutes de les diferents seccions de l'aplicació, etc.

Frameworks PHP

- **Laravel**, un framework relativament recent (va ser creat en 2011).
- **Symfony** , el framework que emprarem en aquest curs.
Creat en 2005.
- **CodeIgniter**, un framework més lleuger que els anteriors.
Va ser creat en 2006.
- **Phalcon**, un altre framework de recent creació (2012).
- **CakePHP**, creat en 2005, és un altre framework similar a CodeIgniter.
- **Laminas**, evolució de Zend Framework creat en 2006.
- ... etc.

Recursos

A l'hora de treballar amb Symfony (versió 5), necessitem:

- Un servidor web que suporti PHP 7.4 o posterior.
- Un servidor de bases de dades en el qual emmagatzemar la informació de les nostres aplicacions. Emprarem un servidor MariaDB/MySQL.
- PHP (versió 7.4 o posterior)
- El propi framework Symfony.
- Un IDE (entorn de desenvolupament) amb el qual editar el codi dels nostres projectes
- Composer

Iniciar el projecte Symfony

Hi ha dues alternatives per a crear projectes Symfony:

```
composer create-project symfony/skeleton project-name
```

Crearà un projecte amb el nom indicat en la carpeta actual, contenint l'estructura mínima, sense llibreries de tercers.

```
composer create-project symfony/website-skeleton project-name
```

Crearà el projecte amb totes els components tradicionals per crear un lloc web complet.


Projecte de prova

Per a començar, anem a crear un projecte anomenat "testing" amb la segona opció. Accedim a la carpeta de treball (podem crear una en /home/alumne/projectes-symfony, per exemple), i escrivim aquest comando des de dins d'aqueixa carpeta:

```
composer create-project symfony/website-skeleton testing
```

Estructura del projecte

En el cas concret d'un projecte Symfony 5, l'estructura queda com segueix:



- bin
- composer.json
- composer.lock
- config
- migrations
- public
- src
- templates
- tests
- translations
- var
- vendor

La consola de Symfony

En la carpeta `bin` hi ha un parell d'arxius executables via PHP.

Un

- `phpunit` , per a fer proves unitàries
- `console` , àmpliament utilitzat per a ajudar-nos a certes tasques, com per exemple crear entitats, engegar un servidor de proves, examinar serveis, etc.

Prova a executar aquest comando en el projecte testing:

```
php bin/console debug:autowiring
```

Introducció a YAML

Format emprat per a la configuració de projectes Symfony.

Més senzill que XML i que JSON, per exemple.

L'estàndard YAML és molt ampli (es pot consultar en el seu web oficial), però per a Symfony només és necessari emprar un subconjunt reduït.

Introducció a YAML

La informació s'estructura en parelles clau, valor. Si la clau o el valor estan compostos per més d'una paraula separades per espais, es tanquen entre cometes dobles (encara que no és habitual).

```
driver: 'pdo_mysql'
```

Introducció a YAML

La jerarquia de la informació s'estableix mitjançant indentacions, però aquestes no han de fer-se amb el tabulador, sinó amb la barra espaciadora (quatre espais per nivell).

```
default_table_options:  
  charset: utf8mb  
  collate: utf8mb4_unicode_ci
```

Introducció a YAML

Pot haver-hi grups d'elements clau: valor, formant arrays. Aquests arrays van entre claudàtors, en el cas d'arrays normals, o entre claus, en el cas d'arrays associatius:

```
rols: [ROLE_USER, ROLE_ADMIN]

users:
  admin: { password: adminpass, rols: [ROLE_ADMIN] }
  user: { password: userpass, rols: [ROLE_USER] }
```

Introducció a YAML

També podem indicar una matriu (*array*) escrivint cada element en una nova línia que comence per -

```
App\  
  resource: '../src/'  
  exclude:  
    - '../src/DependencyInjection/'  
    - '../src/Entity/'  
    - '../src/Kernel.php'  
# exclude: ['../src/DependencyInjection/', '../src/Entity/', '../src/Kernel.php']
```

Introducció a YAML

Els comentaris s'indiquen amb un coixinet al principi de cada línia del comentari.

```
# in-memory users  
users:  
  admin: ...
```

