

# LAB 9 – Programação com o framework Symfony [parte 1]

---

O objectivo deste laboratório é repetir a funcionalidade do portal desenhado no LAB4 mas agora construído com o *framework* **Symfony 4.4** e a *template engine* **Twig**.

## 0. Preliminares

Faça login no servidor daw2<sup>1</sup> (IP 10.10.23.184)

**Download o framework para dentro da pasta “public\_html/LAB9\_10”**

```
a12345@daw2:~$ cd public_html
a12345@daw2:~/public_html$
composer create-project symfony/website-skeleton:"^4.4" LAB9_10
```

O comando cria a pasta LAB9\_10 com a instalação do Symfony 4.4 e todas as livrarias standard.

Note bem: LAB9\_10 **SIM**, Lab9\_10, lab9\_10 **NÃO**

**Altere o ficheiro “.env” para utilizar as credenciais da sua base de dados. Substitua “a12345” pelo seu login e “\*\*\*\*\*” pela password de acesso à sua base de dados**

```
a12345@daw2:~/public_html/LAB9_10$ nano .env
```

ponha um “#” no principio da linha

```
DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13
```

adicione a linha

```
DATABASE_URL=mysql://a12345:*****@127.0.0.1:3306/db_a12345?serverVersion=15.1
```

**Faça uma cópia da base de dados que tem no servidor daw (IP 10.10.23.183) para este servidor**

```
a12345@daw2:~$ mysqldump -u a12345 -p***** -h 10.10.23.183 db_a12345 > myDB.SQL
a12345@daw2:~$ /usr/local/bin/mysql-db
a12345@daw2:~$ mysql -u a12345 -p***** -h localhost db_a12345 < myDB.SQL
```

---

<sup>1</sup> O framework Symfony NÃO funciona no servidor daw (IP 10.10.23.183)

## Crie o controlador “BlogController”

```
a12345@daw2:~/public_html/LAB9_10$ php bin/console make:controller
Choose a name for your controller class (e.g. GrumpyChefController):
> BlogController
```

## Teste a instalação do framework.

A partir do seu browser preferido vá ao seguinte URL

`http://daw.deei.fct.ualg.pt/~a12345/LAB9_10/public/index.php/blog`

Deverá receber uma página web de boas-vindas

## Hello BlogController!

This friendly message is coming from:

- Your controller at [src/Controller/BlogController.php](#)
- Your template at [templates/blog/index.html.twig](#)

---

## (Opcional) URL rewriting

Se quiser evitar escrever “index.php” nos URL, altere a “Rewrite base” no ficheiro `.htaccess`

```
a12345@daw2:~/public_html/LAB9_10/public$ nano .htaccess
```

```
<IfModule mod_rewrite.c>
```

```
    RewriteEngine On
```

```
    # Redirect Trailing Slashes If Not A Folder...
```

```
    RewriteCond %{REQUEST_FILENAME} !-d
```

```
    RewriteRule ^(.*)/$ /$1 [L,R=301]
```

```
    # Handle Front Controller...
```

```
    RewriteCond %{REQUEST_FILENAME} !-d
```

```
    RewriteCond %{REQUEST_FILENAME} !-f
```

```
    RewriteRule ^ index.php [L]
```

```
    # Handle Authorization Header
```

```
    RewriteCond %{HTTP:Authorization} .
```

```
    RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
```

```
</IfModule>
```

## OBSERVAÇÕES

- Sempre que receber uma mensagem de erro informando o acesso negado à pasta "**LAB9\_10/var/cache/dev**",

```
Warning:  
file_put_contents(/users/a12345/public_html/lixo/var/cache/dev/srcApp_KernelDevDebugContain  
erDeprecations.log): failed to open stream: Operation not permitted
```

terá que limpar o "cache" com o comando

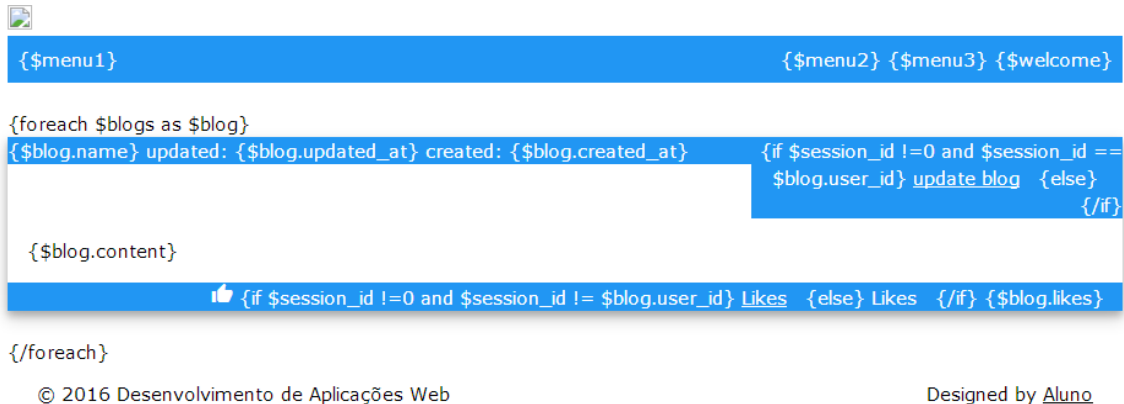
```
a12345@daw2:~/public_html/LAB9_10$ php bin/console cache:clear
```

- Se instalar um componente Symfony utilizando o "composer", poderá ter (nem sempre é necessário) que refazer novamente as permissões na pasta "vendor" com os comandos

```
a12345@daw2:~/public_html$ find LAB9_10/vendor -type d -exec chmod 755 {} \;  
a12345@daw2:~/public_html$ find LAB9_10/vendor -type f -exec chmod 644 {} \;
```

# 1. Desenho do template index\_template.html.twig

Adapte o template index\_template.tpl para o template index\_template.html.twig



que vai ser utilizado para construir a página de rosto do site.

O template index\_template.html.twig deve ser colocado na pasta

public\_html/LAB9\_10/templates/blog

**NOTA:** os recursos utilizados no template (imagens, css, javascript) devem ser colocados em pastas (images, css, js, etc) dentro da pasta “public”

# 2. Construa o controlador BlogController.php na pasta

public\_html/LAB9\_10/src/Controller

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Session\SessionInterface;
use Symfony\Component\PropertyAccess\PropertyAccess;
use Symfony\Component\Validator\Constraints as Assert;
use Symfony\Component\Validator\Validator\ValidatorInterface;
use Symfony\Component\Validator\Context\ExecutionContextInterface;
use App\Controller\Blog_modelController;

class BlogController extends AbstractController
{
    private $blog_model;
```

```

        private $session;
        private $validator;

        public function __construct(Blog_modelController $blog_model, SessionInterface
        $session, ValidatorInterface $validator)
        {
            $this->blog_model = $blog_model;
            $this->session = $session;
            $this->validator = $validator;
        }

```

O código PHP do controlador responsável pela página de rosto deverá encontrar-se na função `index()`

```

/**
 * @Route("/blog", name="blog")
 */
public function index()
{
}

```

### 3. Construção da classe responsável pelo acesso à base de dados.

Construa a classe `Blog_modelController.php` na pasta

`public_html/LAB9_10/src/Controller`

```

<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Doctrine\DBAL\Driver\Connection;

class Blog_modelController extends AbstractController
{
    private $connection;

    public function __construct(Connection $connection)
    {
        $this->connection = $connection;
    }
}

```

Construa a função `get_posts()` responsável pela query à base de dados

```

public function get_posts()
{
}

```

## 4. Se realizou o lab no seu computador pessoal, faça o upload dos ficheiros

- `index_template.html.twig`
- `BlogController.php`
- `Blog_modelController.php`

para os directorios “`templates/blog`” e “`src/Controller`”, respectivamente na pasta “LAB9\_10” no seu site web pessoal

Teste o funcionamento do site no url

[http://daw.deei.fct.ualg.pt/~a12345/LAB9\\_10/public/index.php/blog](http://daw.deei.fct.ualg.pt/~a12345/LAB9_10/public/index.php/blog)

Considere o lab concluído quando obtiver a mesma funcionalidade que foi requerida no LAB4

### IMPORTANTE

- Os recursos locais devem ter URLs relativos: utilize as funções `asset()` e `path()` do Twig!

### REFERÊNCIAS:

- [http://daw.deei.fct.ualg.pt/~a999990/SF\\_exam2/public/blog](http://daw.deei.fct.ualg.pt/~a999990/SF_exam2/public/blog)
- <https://symfony.com/doc/4.4/index.html>
- [http://intranet.deei.fct.ualg.pt/DAW/slides/SF\\_overview.pdf](http://intranet.deei.fct.ualg.pt/DAW/slides/SF_overview.pdf)
- <http://all.deei.fct.ualg.pt/symfony/>
- <https://twig.symfony.com/doc/3.x/>

# ANEXO 1. Estrutura da base de dados

A estrutura da base de dados pode ser consultada em <http://all.deei.fct.ualg.pt/phpMyAdmin>

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(255) default NULL,  
  `email` varchar(255) default NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  `password_digest` varchar(255) default NULL,  
  `remember_digest` varchar(255) default NULL,  
  `admin` tinyint(1) default NULL,  
  `activation_digest` varchar(255) default NULL,  
  `activated` tinyint(1) default NULL,  
  `activated_at` datetime default NULL,  
  `reset_digest` varchar(255) default NULL,  
  `reset_sent_at` datetime default NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `index_users_on_email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `microposts` (  
  `id` int(11) NOT NULL auto_increment,  
  `content` text,  
  `user_id` int(11) default NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  `likes` int(11) NOT NULL DEFAULT 0,  
  PRIMARY KEY (`id`),  
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users`  
  (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```