

////////////////////////////////////Het maken van het project met daarbij een database////////////////////////////////////

//Maak het nieuwe project aan//

composer create-project symfony/website-skeleton (naam van het project) "^4.4"

//Installeer locale symfony server//

composer require symfony/web-server-bundle --dev

//Maak een controller aan//

php bin/console make:controller (DefaultController)

//Maak de default route naar de DefaultController//

ga naar src/controller/DefaultController.php en pas - @Route("/default", name="default") naar - *
@Route("/", name="default")

//(optioneel) zet de bootstrap cdn in templates/base.html.twig tussen het meta blok en de title blok//

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

//Pas de database gegevens aan//

ga naar .env. Zoek naar DATABASE_URL

DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name pas dit aan naar

DATABASE_URL=mysql://root:@127.0.0.1:3306/databaseNaam

(als je zelf een wachtwoord hebt in gesteld op phpmyadmin moet je die invullen op db_password)

//Maak de database (vergeet dit niet in het engels te doen)//

php bin/console doctrine:database:create

//Maak tabellen (vergeet dit niet in het engels te doen en enkelvoudig)//

php bin/console make:entity

//Run de queries//

php bin/console doctrine:schema:update --force

//Maak een crud systeem voor een tabel//

php bin/console make:crud

Vervolgens de naam van de tabel

```
////////////////////////////////////FOSUserBundle////////////////////////////////////  
////////////////////////////////////
```

//Installeer FOSUserBundle//

composer require friendsofsymfony/user-bundle "~2.0"

(de fout ``The child node "db_driver" at path "fos_user" must be configured`` zal getoond worden. Deze kan je negeren)

//Maak de user class//

Maak User.php aan in src/entity en zet vervolgens dit er in.

<?php

namespace App\Entity;

use FOS\UserBundle\Model\User as BaseUser;

use Doctrine\ORM\Mapping as ORM;

/**

* @ORM\Entity

* @ORM\Table(name="fos_user")

*/

class User extends BaseUser

{

/**

* @ORM\Id

* @ORM\Column(type="integer")

* @ORM\GeneratedValue(strategy="AUTO")

*/

protected \$id;

public function __construct()

{

parent::__construct();

// your own logic

}

}

//Update beveiliging//

Ga naar config/packages/security.yaml. Verander vervolgens de code naar dit

security:

encoders:

FOS\UserBundle\Model\UserInterface: bcrypt

role_hierarchy:

ROLE_ADMIN: ROLE_USER

ROLE_SUPER_ADMIN: ROLE_ADMIN

<https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers>

providers:

fos_userbundle:

id: fos_user.user_provider.username_email

firewalls:

dev:

pattern: ^/(_(profiler|wdt)|css|images|js)/

security: false

main:

pattern: ^/

form_login:

provider: fos_userbundle

csrf_token_generator: security.csrf.token_manager

logout: true

anonymous: true

Easy way to control access for large sections of your site

Note: Only the *first* access control that matches will be used

access_control:

- { path: ^/login\$, role: IS_AUTHENTICATED_ANONYMOUSLY }

- { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSLY }

- { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY }

- { path: ^/admin/, role: ROLE_ADMIN }

//Configuratie bestand//

maak fos_user.yaml aan in config/packages. Zet vervolgens dit er in

fos_user:

db_driver: orm # other valid values are 'mongodb' and 'couchdb'

firewall_name: main

user_class: App\Entity\User

```
from_email:
  address: "test@domain.com"
  sender_name: "test@domain.com"
```

//iets met templating ofzo//
ga naar config/packages/framework.yaml. Voeg dit toe

```
templating:
  engines: ['twig', 'php']
```

//lets met routing ofzo//
ga naar config/routes. Maak vervolgens fos_user.yaml aan en zet dit er in.

```
fos_user:
  resource: "@FOSUserBundle/Resources/config/routing/all.xml"
```

//Update je database weer//
php bin/console doctrine:schema:update --force

//Als layout. er voorstaan moet je dit in config/packages/framework.yaml zetten
translator: { fallback: en }

////////////////////////////////////CKeditor////////////////////////////////////
////////////////////////////////////

Shopping cart

////////////////////////////////////
php bin/console make:entity

Product

description - string

price - float

php bin/console make:crud - Product

ProductController -

name = CartIndex in route

base.html.twig -

in <a> de functie zetten

php bin/console doctrine:schema:update --force

bootstrap in config twig.yaml -- horizontal

in CartController

Nieuwe function.Route naar {id}add Verander name naar cart_add

Nieuwe function.Route naar {id}del Verander name naar cart_delete

Nieuwe function.Route naar checkout Verander name naar checkout

index - sessie inlesen en meegeven aan index

add - product id komt binnen en word of opgehoogd of toegevoegd aan sessie

del - product id komt binnen en word verlaagd of verwijderd

checkout - afhandelen van sessie naar dbase

Kijk op Git van Robben naar webshop

in CartController

get session in session variable zetten

getRepository aanpassen - App:Product

Invoice

date - datetime

user (foreign) - int

Maak crud hiervan

Rows

product (foreign) - relation - Product -ManyToOne - no - no

invoice (foreign) - relation - Invoice -ManyToOne - no - no

rows - float - no

Maak crud hiervan

Alle screenshots zijn voor CartController