



Ministerul Educației, Culturii și Cercetării a Republicii Moldova

Universitatea de Stat a Moldovei

Facultatea Matematică și Informatică

Raport

Lucrare de laborator Nr.5

la disciplina Framework-uri pentru Dezvoltarea de Aplicații Web

Tema:” Baze de date și doctrină ORM”

A efectuat : Batîr Ana

A verificat : Bodrug Svetlana

Chișinău 2020

Instalarea Doctrinei

Mai întâi, instalez suport Doctrine prin pachetul orm Symfony, precum și MakerBundle, care mă va ajuta să genereze un anumit cod:

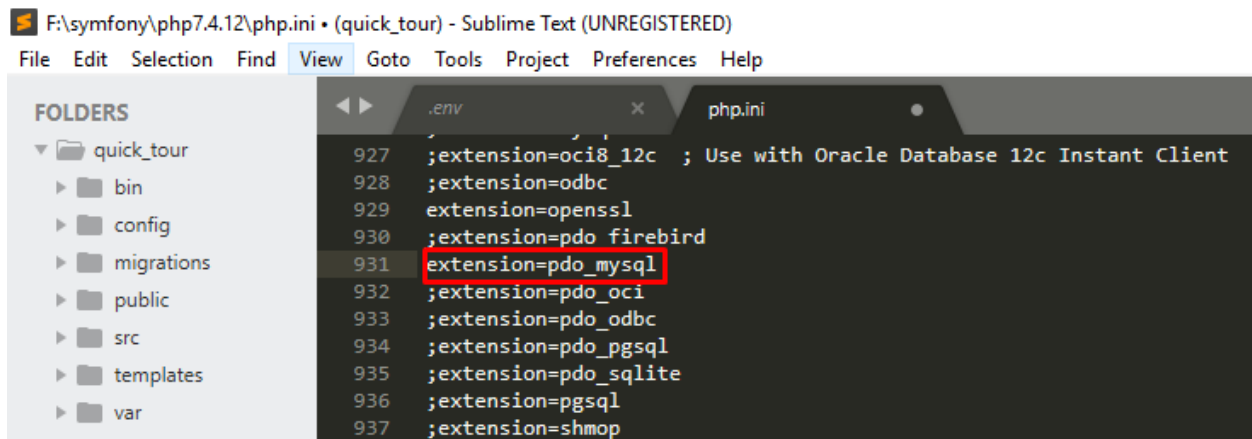
```
F:\symfony\Lab5\quick_tour>composer require symfony/orm-pack
Using version ^2.0 for symfony/orm-pack
./composer.json has been updated
Running composer update symfony/orm-pack
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking symfony/orm-pack (v2.0.0)
Writing lock file
```

```
F:\symfony\Lab5\quick_tour>composer require --dev symfony/maker-bundle
Using version ^1.23 for symfony/maker-bundle
./composer.json has been updated
Running composer update symfony/maker-bundle
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking nikic/php-parser (v4.10.2)
  - Locking symfony/maker-bundle (v1.23.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
```

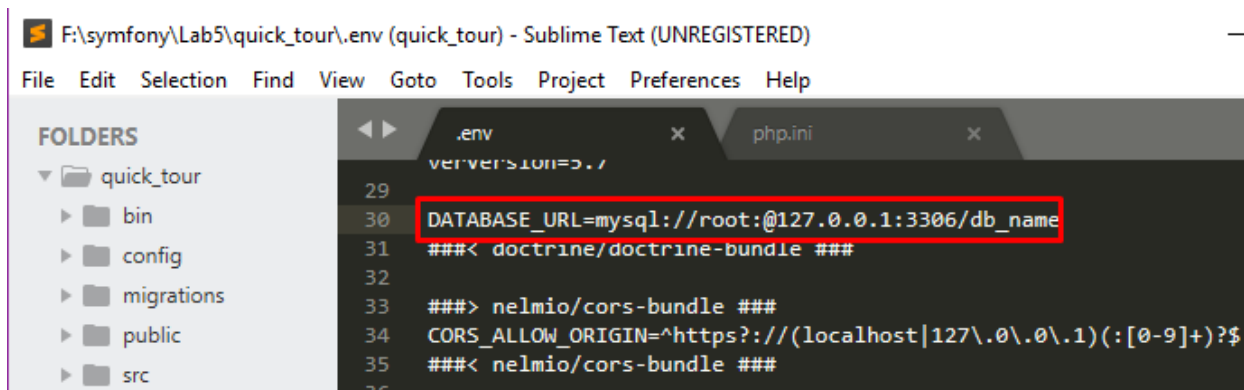
Configurarea bazei de date

Informațiile de conectare la baza de date sunt stocate ca o variabilă de mediu numită DATABASE_URL. Pentru dezvoltare, pot găsi și personaliza acest lucru în interiorul .env.

Activare MySQL în fișierul php.ini .



Acum că parametrii de conexiune sunt configurați, Doctrine poate crea baza de date db_name:



The screenshot shows the Sublime Text editor with the file `F:\symfony\Lab5\quick_tour\.env` open. The file contains the following configuration:

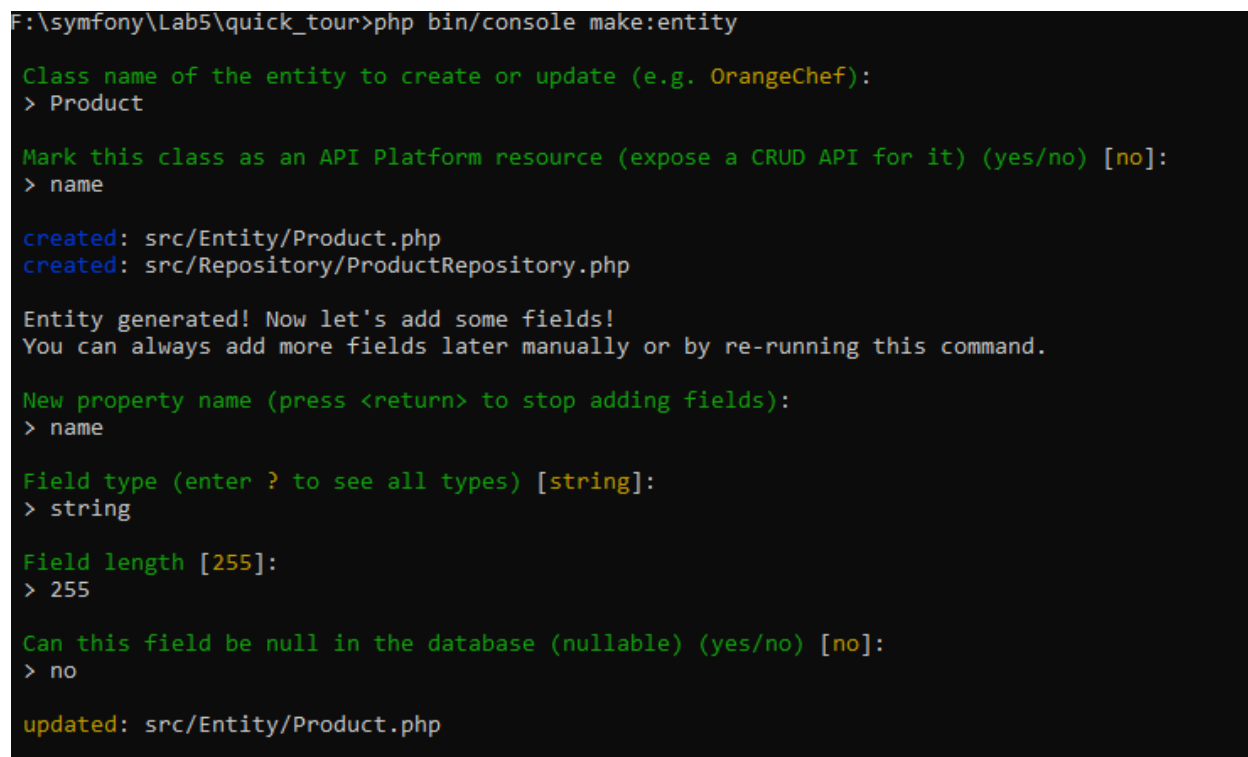
```
verversion=3.7
29
30 DATABASE_URL=mysql://root:@127.0.0.1:3306/db_name
31 ###< doctrine/doctrine-bundle ###
32
33 ###> nelmio/cors-bundle ###
34 CORS_ALLOW_ORIGIN=https://(localhost|127.0.0.1|10\.\.\.1)(:[0-9]+)?$
35 ###< nelmio/cors-bundle ###
36
```

```
F:\symfony\Lab5\quick_tour>php bin/console doctrine:database:create
Created database `db_name` for connection named default
```

Crearea unei clase de entitate

Să presupunem că creem o aplicație în care trebuie afișate produse. Fără să vă gândiți la Doctrine sau baze de date, știm deja că am nevoie de un obiect `Produs` pentru a reprezenta aceste produse.

Pot utiliza comanda `make:entity` pentru a crea această clasă și orice câmpuri de care am nevoie. Comanda va pune câteva întrebări - răspund așa cum se face mai jos:



```
F:\symfony\Lab5\quick_tour>php bin/console make:entity

Class name of the entity to create or update (e.g. OrangeChef):
> Product

Mark this class as an API Platform resource (expose a CRUD API for it) (yes/no) [no]:
> name

created: src/Entity/Product.php
created: src/Repository/ProductRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> name

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
> 255

Can this field be null in the database (nullable) (yes/no) [no]:
> no

updated: src/Entity/Product.php
```

Iuhu! Acum am un nou fișier src / Entity / Product.php:

```
Add another property? Enter the property name (or press <return> to stop adding fields):  
> price  
  
Field type (enter ? to see all types) [string]:  
> integer  
  
Can this field be null in the database (nullable) (yes/no) [no]:  
> no  
  
updated: src/Entity/Product.php  
  
Add another property? Enter the property name (or press <return> to stop adding fields):  
>
```

Success!

Next: When you're ready, create a migration with `php bin/console make:migration`

F:\symfony\Lab5\quick_tour>_

```

1 |<?php
2
3 namespace App\Entity;
4
5 use App\Repository\ProductRepository;
6 use Doctrine\ORM\Mapping as ORM;
7
8 /**
9  * @ORM\Entity(repositoryClass=ProductRepository::class)
10  */
11 class Product
12 {
13     /**
14      * @ORM\Id
15      * @ORM\GeneratedValue
16      * @ORM\Column(type="integer")
17      */
18     private $id;
19
20     /**
21      * @ORM\Column(type="string", length=255)
22      */
23     private $name;
24
25     /**
26      * @ORM\Column(type="integer")
27      */
28     private $price;
29
30     public function getId(): ?int
31     {
32         return $this->id;
33     }
34
35     public function getName(): ?string
36     {
37         return $this->name;
38     }
39
40     public function setName(string $name): self
41     {
42         $this->name = $name;

```

Migrații: crearea tabelelor / schemei bazei de date

Clasa de produs este complet configurată și gata de salvare într-un tabel de produse. Dacă tocmai am definit această clasă, baza de date nu are încă tabelul de produse. Pentru a-l adăuga, pot utiliza DoctrineMigrationsBundle, care este deja instalat:

```
F:\symfony\Lab5\quick_tour>php bin/console make:migration

Success!

Next: Review the new migration "migrations/Version20201115235601.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

F:\symfony\Lab5\quick_tour>
```

Dacă deschid acest fișier, acesta conține codul SQL necesar pentru actualizarea bazei de date!
Pentru a rula acel SQL, execut migrațiile:

```
F:\symfony\Lab5\quick_tour>php bin/console doctrine:migrations:migrate

WARNING! You are about to execute a database migration that could result in schema changes
and data loss. Are you sure you wish to continue? (yes/no) [yes]:
> yes

[notice] Migrating up to DoctrineMigrations\Version20201115235601
[notice] finished in 156.5ms, used 20M memory, 1 migrations executed, 1 sql queries

F:\symfony\Lab5\quick_tour>
```

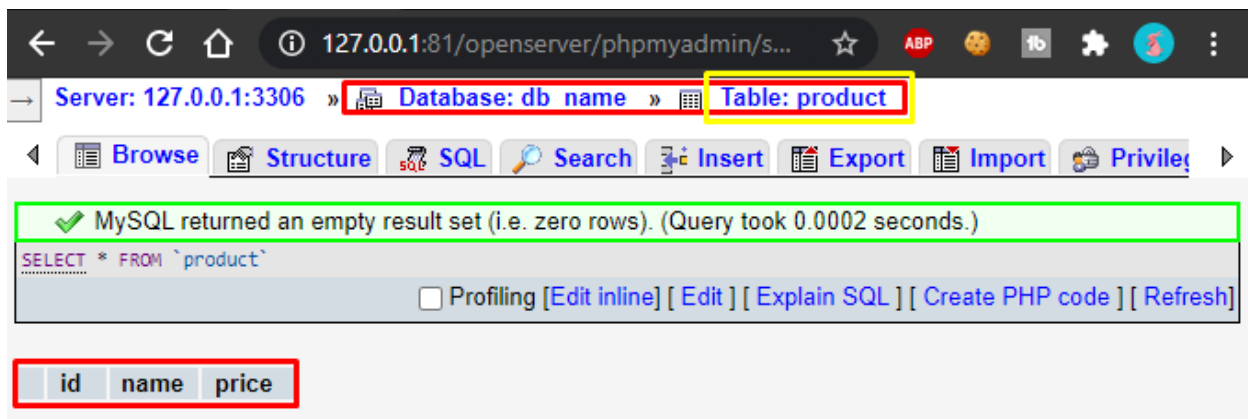
Server: 127.0.0.1:3306 » Database: db_name

Structure SQL Search Query Export Import Operations Pri

Filters
Containing the word:

Table	Action	Rows	Type	Collation
<input type="checkbox"/> doctrine_migration_versions	★ [icons]	4	InnoDB	utf8_unicode_c
<input type="checkbox"/> product	★ [icons]	0	InnoDB	utf8mb4_unico
2 tables	Sum	4	InnoDB	utf8_general_c

Check all With selected:



Migrații și adăugarea mai multor câmpuri

Dar dacă trebuie să adaug o nouă proprietate de câmp la produs, cum ar fi o descriere? Pot edita clasa pentru a adăuga noua proprietate. Dar, pot utiliza și make: entity din nou:

```
F:\symfony\Lab5\quick_tour>php bin/console make:entity

Class name of the entity to create or update (e.g. OrangeElephant):
> Product

Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):
> description

Field type (enter ? to see all types) [string]:
> text

Can this field be null in the database (nullable) (yes/no) [no]:
> no

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!

Next: When you're ready, create a migration with php bin/console make:migration

F:\symfony\Lab5\quick_tour>
```

```
.src\Entity\Product.php (quick_tour) - Sublime Text (UNREGISTERED) —
ew Goto Tools Project Preferences Help

Product.php x
29
30 /**
31  * @ORM\Column(type="text")
32  */
33 private $description;
34
35 public function getId(): ?int
36 {
37     return $this->id;
38 }
```

Noua proprietate este mapată, dar nu există încă în tabelul de produse. Nici o problema! Generez o nouă migrație:

```
F:\symfony\Lab5\quick_tour>php bin/console make:migration

Success!

Next: Review the new migration "migrations/Version20201116000113.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

F:\symfony\Lab5\quick_tour>
```

Sistemul de migrație este inteligent. Compară toate entitățile cu starea actuală a bazei de date și generează SQL-ul necesar pentru a le sincroniza! Ca și înainte, execut migrările:

```
F:\symfony\Lab5\quick_tour>php bin/console doctrine:migrations:migrate

WARNING! You are about to execute a database migration that could result in schema changes
and data loss. Are you sure you wish to continue? (yes/no) [yes]:
> yes

[notice] Migrating up to DoctrineMigrations\Version20201116000113
[notice] finished in 63.8ms, used 20M memory, 1 migrations executed, 1 sql queries

F:\symfony\Lab5\quick_tour>
```


The screenshot shows the phpMyAdmin web interface in a browser. The address bar shows the URL `127.0.0.1:81/openserver/phpmyadmin/s...`. The breadcrumb navigation at the top indicates the path: `Server: 127.0.0.1:3306 » Database: db_name » Table: product`. Below this, a toolbar contains buttons for `Browse`, `Structure`, `SQL`, `Search`, `Insert`, `Export`, `Import`, and `Privileges`. A green message box states: `MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)`. Below the message, the SQL query `SELECT * FROM `product`` is displayed. To the right of the query are links for `Profiling`, `Edit inline`, `Edit`, `Explain SQL`, `Create PHP code`, and `Refresh`. At the bottom, a table structure is shown with four columns: `id`, `name`, `price`, and `description`.

Obiecte persistente în baza de date

Este timpul să salvez un obiect `Produs` în baza de date! Să creez un controler nou pentru a experimenta:

```
F:\symfony\Lab5\quick_tour>php bin/console make:controller ProductController
created: src/Controller/ProductController.php
created: templates/product/index.html.twig

Success!

Next: Open your new controller class and add some pages!
F:\symfony\Lab5\quick_tour>
```

În interiorul controlerului, pot crea un nou obiect de produs, pot seta date pe acesta și îl pot salva:

```
\src\Controller\ProductController.php (quick_tour) - Sublime Text ...
View Goto Tools Project Preferences Help

Product.php x ProductController.php x
3 namespace App\Controller;
4
5 use Symfony\Bundle\FrameworkBundle\Controller\
  AbstractController;
6 use Symfony\Component\Routing\Annotation\Route;
7 use App\Entity\Product;
8 use Doctrine\ORM\EntityManagerInterface;
9 use Symfony\Component\HttpFoundation\Response;
10
11 class ProductController extends AbstractController
12 {
13     /**
14      * @Route("/product", name="product")
15      */
16     public function index(): Response
17     {
18         return $this->render('product/index.html.twig', [
19             'controller_name' => 'ProductController',
20         ]);
21     }
22
23     /**
24      * @Route("/product", name="create_product")
25      */
26     public function createProduct(): Response
27     {
28         // you can fetch the EntityManager via
29         // $this->getDoctrine()
30         // or you can add an argument to the action:
31         // createProduct(EntityManagerInterface
32         // $entityManager)
33         $entityManager = $this->getDoctrine()->getManager(
34             );
35
36         $product = new Product();
37         $product->setName('Keyboard');
38         $product->setPrice(1999);
39         $product->setDescription('Ergonomic and stylish!')
40         ;
41
42         // tell Doctrine you want to (eventually) save
43         // the Product (no queries yet)
44     }
45 }
```

REZULTAT:

← → ↻ 🏠 ⓘ localhost:8000/product

Saved new product with id 1

← → ↻ 🏠 ⓘ localhost:8000/product

Saved new product with id 2

→ Server: 127.0.0.1:3306 » Database: db_name » Table: product

◀ Browse Structure SQL Search Insert Export Import Privileges ▶

✓ Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT * FROM `product`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 50 ▼

+ Options

				id	name	price	description
<input type="checkbox"/>				1	Keyboard	1999	Ergonomic and stylish!
<input type="checkbox"/>				2	Keyboard	1999	Ergonomic and stylish!

↑ ☐ Check all With selected:

Validarea obiectelor

Validatorul Symfony reutilizează metadatele Doctrine pentru a efectua câteva sarcini de validare de bază:

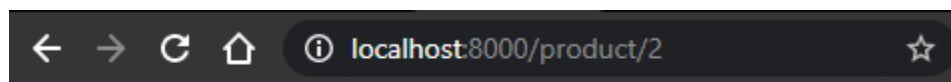
```
src\Controller\ProductController.php (quick_tour) - Sublime Text ...  
ew Goto Tools Project Preferences Help  
Product.php x ProductController.php x index.html.twig x  
9 use Symfony\Component\HttpFoundation\Response;  
10 use Symfony\Component\Validator\Validator\    
    ValidatorInterface;  
11  
12 class ProductController extends AbstractController  
13 {  
14     /**  
15      * @Route("/product1", name="product")  
16      */  
17     public function index(): Response  
18     {  
19         return $this->render('product/index.html.twig', [  
20             'controller_name' => 'ProductController',  
21         ]);  
22     }  
23  
24     /**  
25      * @Route("/product", name="create_product")  
26      */  
27     public function create(ProductValidatorInterface $    
        validator): Response  
28     {  
29         // you can fetch the EntityManager via  
30         $this->getDoctrine()  
31         // or you can add an argument to the action:  
32         createProduct(EntityManagerInterface  
33         $entityManager)  
34         $entityManager = $this->getDoctrine()->getManager(  
35             );  
36  
37         $product = new Product();  
38         $product->setName('Keyboard');  
39         $product->setPrice(1999);  
40         $product->setDescription('Ergonomic and stylish!')  
41         :  
42         $errors = $validator->validate($product);  
43         if (count($errors) > 0) {  
44             return new Response((string) $errors, 400);  
45         }  
46  
47         // tell Doctrine you want to (eventually) save  
48         the Product (no queries yet)  
49         $entityManager->persist($product);  
50     }  
51 }
```

Preluarea obiectelor din baza de date

Extragerea unui obiect din baza de date este și mai ușoară. Să presupunem că pot merge la /product / 1 pentru a vedea noul produs:

```
(src\Controller\ProductController.php (quick_tour) - Sublime Text ...  
ew Goto Tools Project Preferences Help  
Product.php x ProductController.php x index.html.twig x  
50 /**  
51  * @Route("/product/{id}", name="product_show")  
52  */  
53 public function show($id)  
54 {  
55     $product = $this->getDoctrine()  
56         ->getRepository(Product::class)  
57         ->find($id);  
58  
59     if (!$product) {  
60         throw $this->createNotFoundException(  
61             'No product found for id '.$id  
62         );  
63     }  
64  
65     return new Response('Check out this great product: '.$  
66         product->getName());  
67  
68     // or render a template  
69     // in the template, print things with {{ product.name  
70     }}  
71     // return $this->render('product/show.html.twig',  
72     ['product' => $product]);  
73 }
```

REZULTAT:



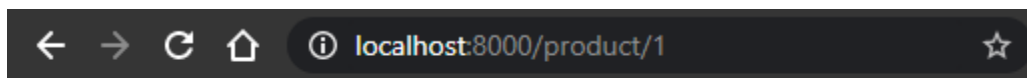
Actualizarea unui obiect

După ce am preluat un obiect din Doctrine, interacționez cu el la fel ca cu orice model PHP:

```
src\Controller\ProductController.php (quick_tour) - Sublime Text ...
ew Goto Tools Project Preferences Help

Product.php x ProductController.php x index.html.twig x
72 // return $this->render('product/show.html.twig',
   ['product' => $product]);
73 }
74 /**
75  * @Route("/product/edit/{id}")
76  */
77 public function update($id)
78 {
79     $entityManager = $this->getDoctrine()->getManager();
80     $product = $entityManager->getRepository(Product::
        class)->find($id);
81
82     if (!$product) {
83         throw $this->createNotFoundException(
84             'No product found for id '.$id
85         );
86     }
87
88     $product->setName('New product name!');
89     $entityManager->flush();
90
91     return $this->redirectToRoute('product_show', [
92         'id' => $product->getId()
93     ]);
94 }
95
96 }
97
```

REZULTAT:



Check out this great product: New product name!