



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 : Pistol Adrian

A verificat : Bodrug Svetlana

Chișinău 2020

Cuprins

Instalarea Doctrinei	3
Configurarea bazei de date	3
Crearea unei clase de entitate	4
Migrații: crearea tabelelor / schemei bazei de date	6
Migrații și adăugarea mai multor câmpuri.....	8
Obiecte persistente în baza de date	9
Rezultat:.....	10
Validarea obiectelor.....	11
Preluarea obiectelor din baza de date.....	11
Rezultat:.....	12
Actualizarea unui obiect.....	12
Rezultat:.....	12

Instalarea Doctrinei

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

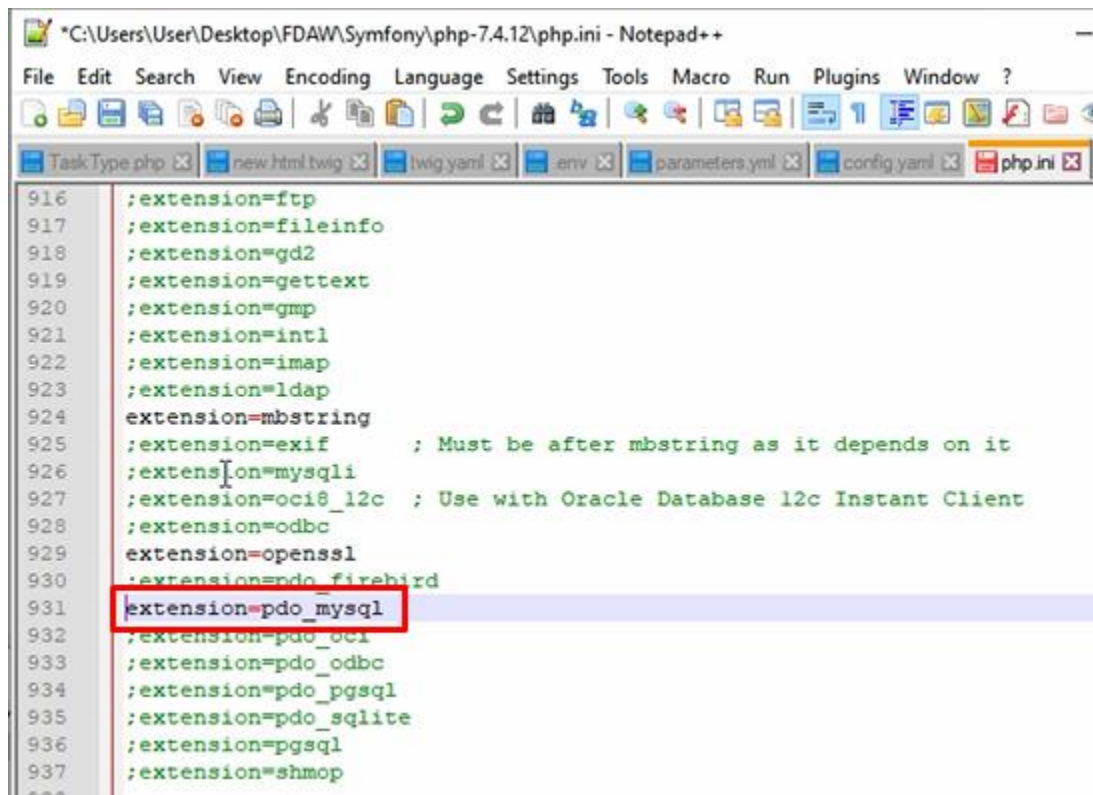
```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> composer require symfony/orm-pack
Using version ^2.0 for symfony/orm-pack
./composer.json has been updated
Running composer update symfony/orm-pack
Loading composer repositories with package information
Updating dependencies
```

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> 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
```


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:



```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name\env - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
twig.yml env parameters.yml config.yml php.ini doctrine.yml Product.php
20 #TRUSTED_HOSTS='^(localhost|example\.com)$'
21 ###< symfony/framework-bundle ###
22
23 ###> symfony/mailer ###
24 # MAILER_DSN=smtp://localhost
25 ###< symfony/mailer ###
26
27 ###> doctrine/doctrine-bundle ###
28 # Format described at
29 https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
30 # For an SQLite database, use: "sqlite:///kernel.project_dir/var/data.db"
31 # For a PostgreSQL database, use:
32 "postgres://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=11&charset=utf8"
33
34 # IMPORTANT: You MUST configure your server version, either here or in
35 config/packages/doctrine.yml
36 #DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7
37
38 DATABASE_URL=mysql://root@127.0.0.1:3306/db_name
39
40 ###< doctrine/doctrine-bundle ###
41
42 ###> nelmio/cors-bundle ###
43 CORS_ALLOW_ORIGIN=http://localhost:127.0.0.1:5081+126
```

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> php bin/console doctrine:database:create
Created database `db name` for connection named default
```

Crearea unei clase de entitate

Să presupunem că creez o aplicație în care trebuie afișate produse. Fără să vă gândiți la Doctrină 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:

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name>php bin/console make:entity

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

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`

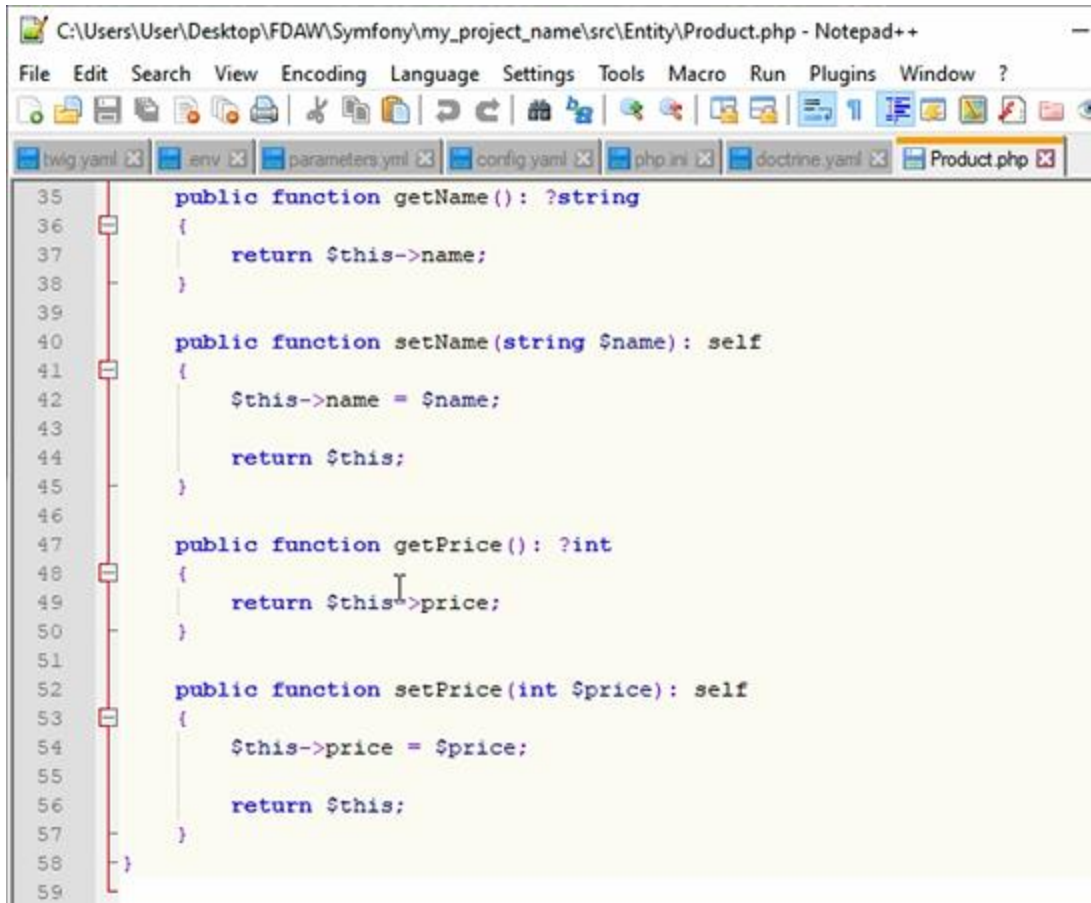
C:\Users\User\Desktop\FDAW\Symfony\my_project_name>

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

```

C:\Users\User\Desktop\FDAW\Symfony\my_project_name\src\Entity\Product.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
twig.yaml x env x parameters.yaml x config.yaml x php.ini x doctrine.yaml x Product.php x
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     /**

```



```
35 public function getName(): ?string
36 {
37     return $this->name;
38 }
39
40 public function setName(string $name): self
41 {
42     $this->name = $name;
43
44     return $this;
45 }
46
47 public function getPrice(): ?int
48 {
49     return $this->price;
50 }
51
52 public function setPrice(int $price): self
53 {
54     $this->price = $price;
55
56     return $this;
57 }
58 }
59
```

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:



```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> php bin/console make:migration

Success!

Next: Review the new migration "migrations/Version20201111225523.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

C:\Users\User\Desktop\FDAW\Symfony\my_project_name>
```

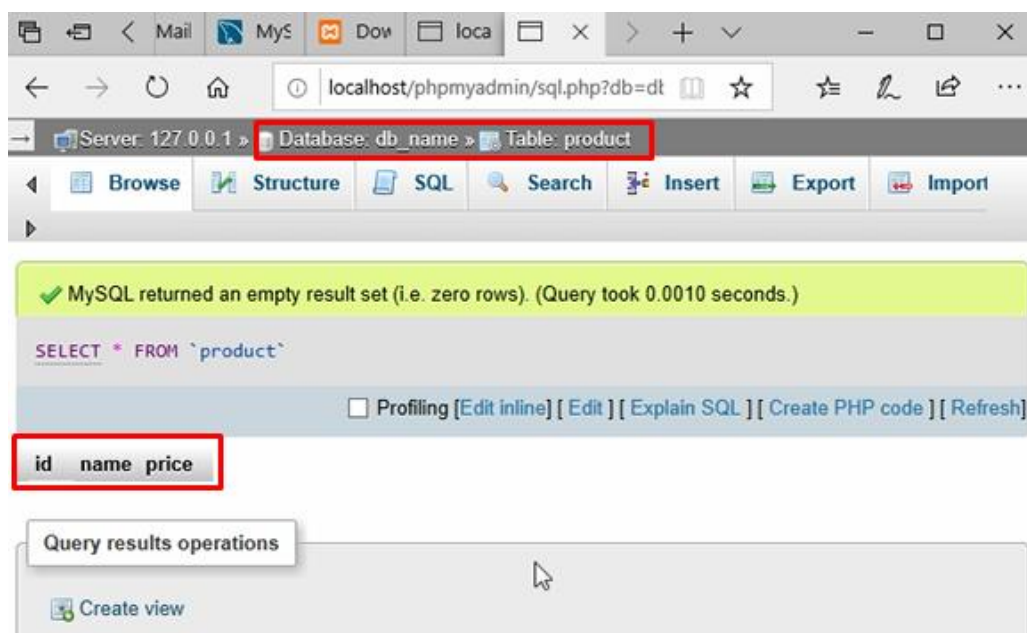
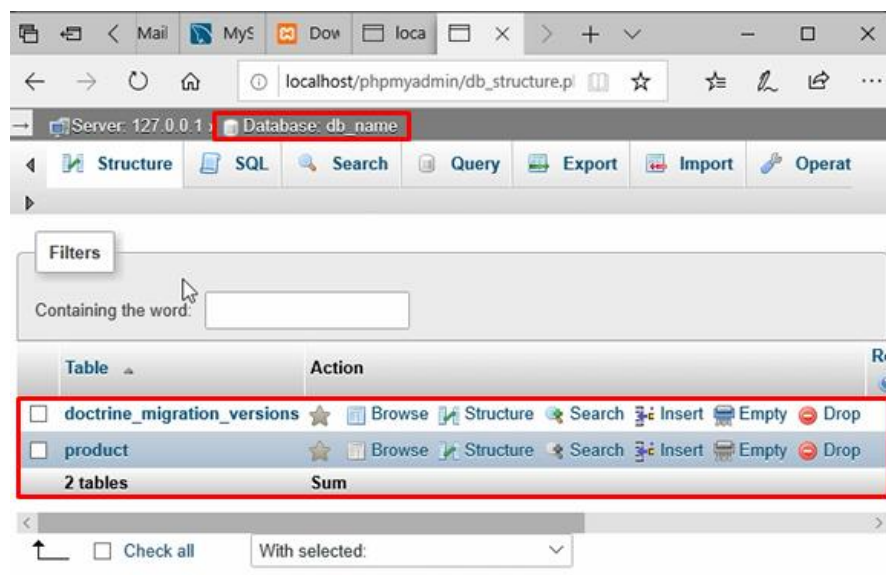

Dacă deschid acest fișier, acesta conține codul SQL necesar pentru actualizarea bazei de date!
Pentru a rula acel SQL, execut migrările:

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> 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\Version20201111225523
[notice] finished in 103.5ms, used 24M memory, 1 migrations executed, 1 sql queries

C:\Users\User\Desktop\FDAW\Symfony\my_project_name>
```



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:

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name>php bin/console make:entity

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

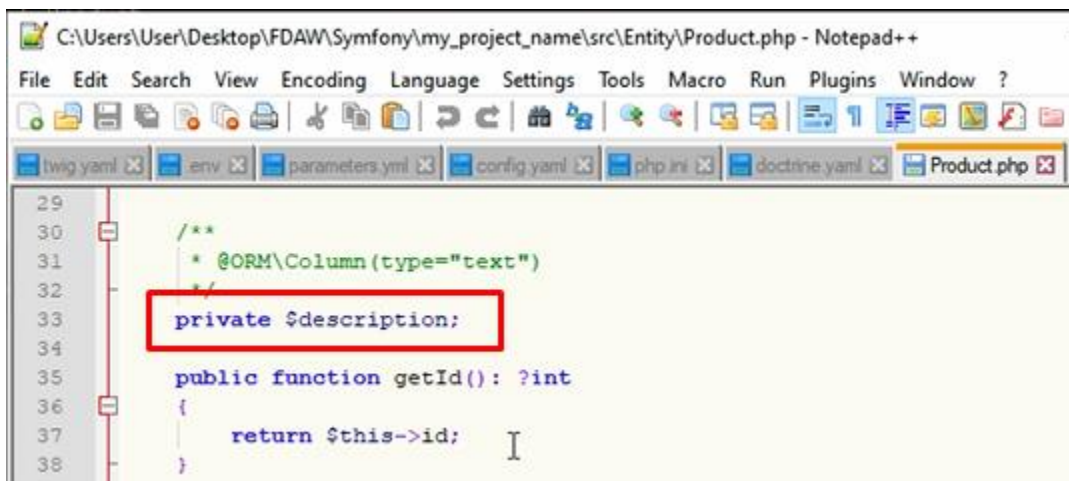
Your entity already exists! So let's add some new fields!

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
```



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

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> php bin/console make:migration

Success!

Next: Review the new migration "migrations/Version20201111230020.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

C:\Users\User\Desktop\FDAW\Symfony\my_project_name>
```

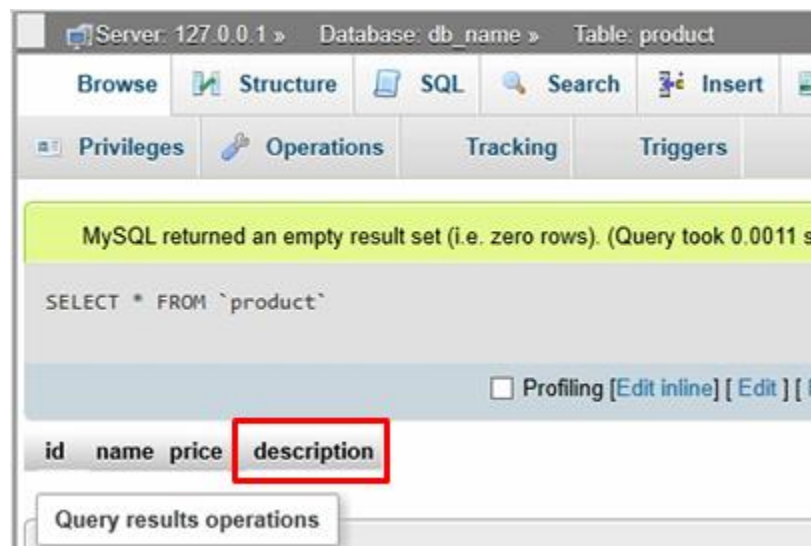

Sistemul de migrație este inteligent. Compară toate entitățile cu starea actuală a bazei de date și generează SQL-ul necesar pentru a le sincroniza! Ca și înainte, execut migrările:

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> 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\Version20201111230020
[notice] finished in 97.4ms, used 24M memory, 1 migrations executed, 1 sql queries

C:\Users\User\Desktop\FDAW\Symfony\my_project_name>
```



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:

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name> 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!

C:\Users\User\Desktop\FDAW\Symfony\my_project_name>
```

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

```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name\src\Controller\ProductController.php - Notepad...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
.env parameters.yml config.yml php.ini doctrine.yml Product.php ProductController.php
14  * @Route("/product2", 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 $this->getDoctrine()
29      // or you can add an argument to the action:
30      createProduct(EntityManagerInterface $entityManager)
31      $entityManager = $this->getDoctrine()->getManager();
32
33      $product = new Product();
34      $product->setName('Keyboard');
35      $product->setPrice(1999);
36      $product->setDescription('Ergonomic and stylish!');
37
38      // tell Doctrine you want to (eventually) save the Product (no queries
```

Rezultat:



Saved new product with id 1

Server: 127.0.0.1 » Database: db_name » Table: product

Browse Structure SQL Search Insert Export Import Pr

SELECT * FROM `product`

☐ Show all | Number of rows: 25 | Filter rows: Search this table

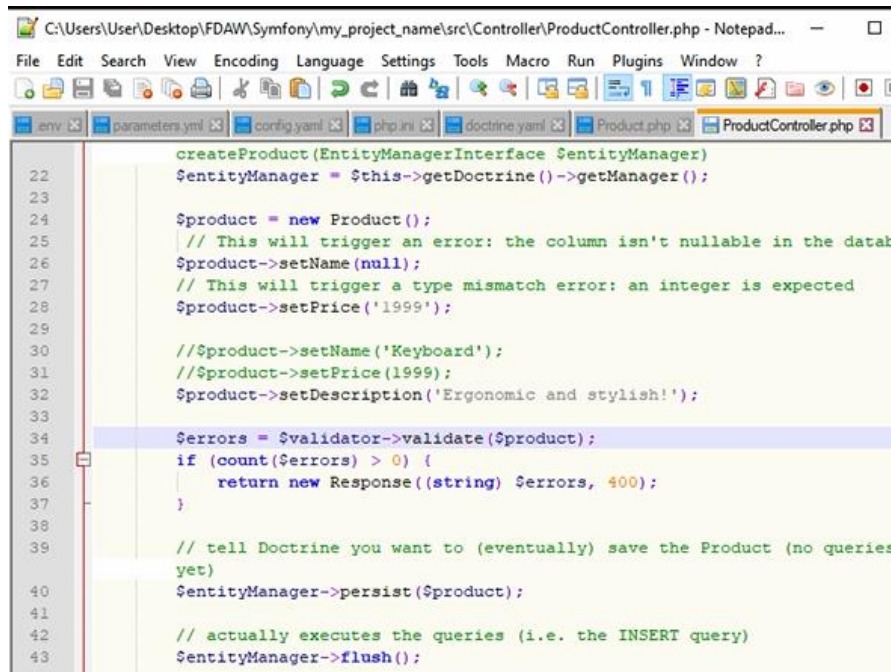
	id	name	price	description
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Keyboard	1999	Ergonomic and stylish!

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Validarea obiectelor

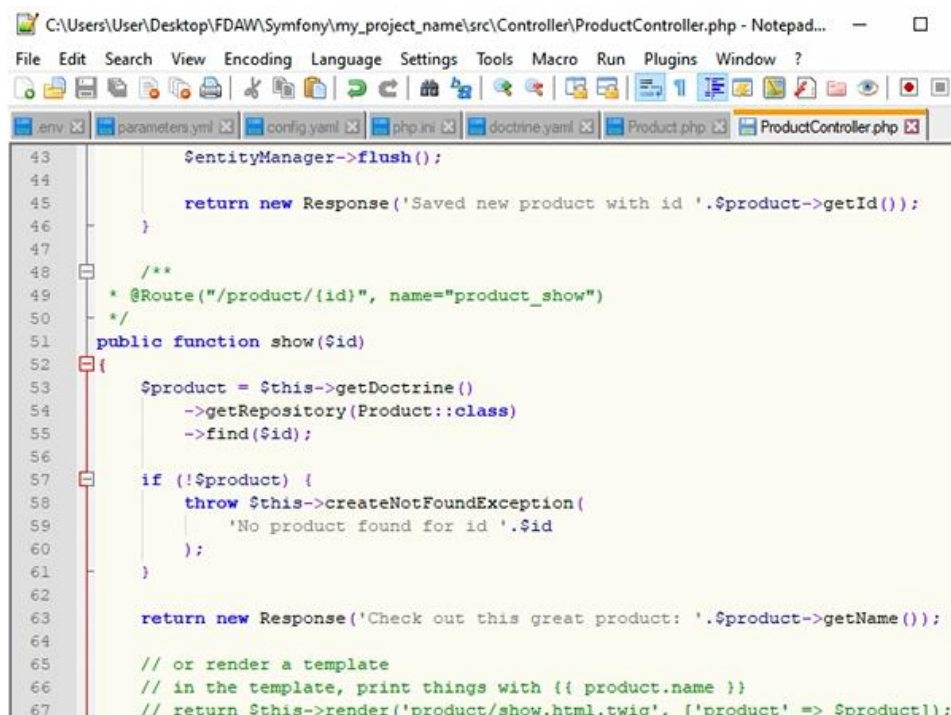
Validatorul Symfony reutilizează metadatele Doctrine pentru a efectua câteva sarcini de validare de bază:



```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name\src\Controller\ProductController.php - Notepad...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
.env parameters.yml config.yml php.ini doctrine.yml Product.php ProductController.php
22 createProduct(EntityManagerInterface $entityManager)
23 $entityManager = $this->getDoctrine()->getManager();
24
25 $product = new Product();
26 // This will trigger an error: the column isn't nullable in the database
27 $product->setName(null);
28 // This will trigger a type mismatch error: an integer is expected
29 $product->setPrice('1999');
30
31 // $product->setName('Keyboard');
32 // $product->setPrice(1999);
33 $product->setDescription('Ergonomic and stylish!');
34
35 $errors = $validator->validate($product);
36 if (count($errors) > 0) {
37     return new Response((string) $errors, 400);
38 }
39
40 // tell Doctrine you want to (eventually) save the Product (no queries yet)
41 $entityManager->persist($product);
42
43 // actually executes the queries (i.e. the INSERT query)
44 $entityManager->flush();
```

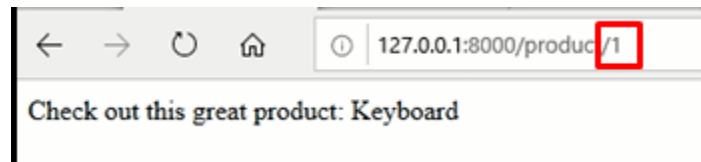
Preluarea obiectelor din baza de date

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



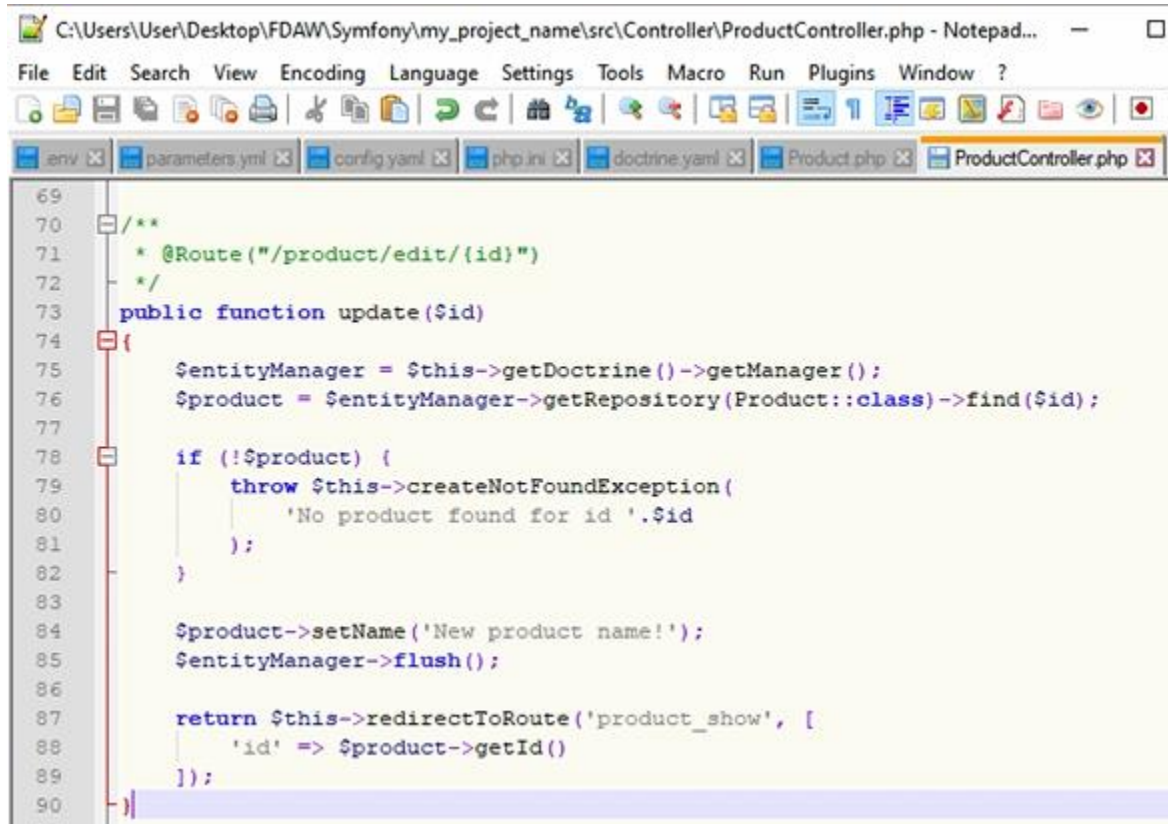
```
C:\Users\User\Desktop\FDAW\Symfony\my_project_name\src\Controller\ProductController.php - Notepad...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
.env parameters.yml config.yml php.ini doctrine.yml Product.php ProductController.php
43 $entityManager->flush();
44
45 return new Response('Saved new product with id '.$product->getId());
46 }
47
48 /**
49  * @Route("/product/{id}", name="product_show")
50  */
51 public function show($id)
52 {
53     $product = $this->getDoctrine()
54         ->getRepository(Product::class)
55         ->find($id);
56
57     if (!$product) {
58         throw $this->createNotFoundException(
59             'No product found for id '.$id
60         );
61     }
62
63     return new Response('Check out this great product: '.$product->getName());
64
65     // or render a template
66     // in the template, print things with {{ product.name }}
67     // return $this->render('product/show.html.twig', ['product' => $product]);
```

Rezultat:



Actualizarea unui obiect

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



Rezultat:

