Custom Repositories and DBAL

Table Relations, Complex Queries, Custom Repository Methods

greenwich.edu.vn

Table of Contents

DOCTRINE

Doctrine DBAL, Doctrine ORM

UNIVERSITY of GREENWICH

Alliance with FPT Education

Create an Entity

```
$ php bin/console make:entity

Class name of the entity to create or update:
> Product

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

New property name (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

New property name (press <return> to stop adding fields):
>
(press enter again to finish)
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Doctrine ORM

- Doctrine Entity

```php
// src/Entity/Product.php
namespace App\Entity;

use App\Repository\ProductRepository;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass=ProductRepository::class)
 */
class Product
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $name;

    /**
     * @ORM\Column(type="integer")
     */
    private $price;

    public function getId(): ?int
    {
        return $this->id;
    }

    // ... getter and setter methods
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Migrations & Adding more Fields

```
$ php bin/console make:entity

Class name of the entity to create or update
> Product

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

New property name (press <return> to stop adding fields):
>
(press enter again to finish)
```

```
// src/Entity/Product.php
// ...

class Product
{
    // ...

+     /**
+      * @ORM\Column(type="text")
+      */
+     private $description;

    // getDescription() & setDescription() were also added
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Query the database directly:

```
> php bin/console doctrine:query:sql 'SELECT * FROM product'

# on Windows systems not using Powershell, run this command instead:
# php bin/console doctrine:query:sql "SELECT * FROM product"
```

UNIVERSITY of GREENWICH

Alliance with FPT Education

Fetching Objects from the Database

```php
// src/Controller/ProductController.php
namespace App\Controller;

use App\Entity\Product;
use Symfony\Component\HttpFoundation\Response;
// ...

class ProductController extends AbstractController
{
    /**
     * @Route("/product/{id}", name="product_show")
     */
    public function show(int $id): Response
    {
        $product = $this->getDoctrine()
            ->getRepository(Product::class)
            ->find($id);

        if (!$product) {
            throw $this->createNotFoundException(
                'No product found for id '.$id
            );
        }

        return new Response('Check out this great product: '.$product->getName());

        // or render a template
        // in the template, print things with {{ product.name }}
        // return $this->render('product/show.html.twig', ['product' => $product]);
    }
}
```

UNIVERSITY of GREENWICH

Alliance with FPT Education

Updating an Object

```php
// src/Controller/ProductController.php
namespace App\Controller;

use App\Entity\Product;
use App\Repository\ProductRepository;
use Symfony\Component\HttpFoundation\Response;
// ...

class ProductController extends AbstractController
{
    /**
     * @Route("/product/edit/{id}")
     */
    public function update(int $id): Response
    {
        $entityManager = $this->getDoctrine()->getManager();
        $product = $entityManager->getRepository(Product::class)->find($id);

        if (!$product) {
            throw $this->createNotFoundException(
                'No product found for id '.$id
            );
        }

        $product->setName('New product name!');
        $entityManager->flush();

        return $this->redirectToRoute('product_show', [
            'id' => $product->getId()
        ]);
    }
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Deleting an Object

```
$entityManager->remove($product);
$entityManager->flush();
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Doctrine ORM

- Doctrine Entity - Table Relations
  - @OneToOne - each row in one table is linked to exactly one row in another table

```
/**
 * @ORM\OneToOne(targetEntity="Contact")
 * @ORM\JoinColumn(name="contact_id", referencedColumnName="id")
 */
private $contact;
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Doctrine ORM

- Doctrine Entity - Table Relations
  - @OneToMany / @ManyToOne - each row in one table is linked to many rows in another table

```
/**
 * @ORM\ManyToOne(targetEntity=Supplier::class, inversedBy="parts")
 */
private $supplier;
```

Doctrine ORM

- Doctrine Entity - Join Tables
  - @ManyToMany - one or more rows in a table can be related to 0, 1 or many rows in another table
  - Many-to-many relation is implemented with join table / mapping table

car table	
car_id	primary key

mapping table	
car_id	
part_id	

part table	
part_id	primary key

UNIVERSITY of GREENWICH
Alliance with FPT Education

EntityManager

- Doctrine Query Language – DQL
- Very similar to SQL
- With DQL we can update, delete, select entities, but not persist
- With DQL we select objects instead of table rows

```php
/**
 * @Route("/car_action_dsc", name="car_action_dsc")
 */
public function carActionDsc() {
    $em = $this->getDoctrine()->getManager();

    $query = $em->createQuery(
        dql: "
            SELECT c
            FROM App\Entity\Car c
            ORDER BY c.travelledDistance DESC
        ");

    $result = $query->getResult();

    return $this->render( view: 'car/index.html.twig', array(
        'cars' => $result,
    ));
}
```

UNIVERSITY of GREENWICH

Alliance with FPT Education

Doctrine ORM

```php
/**
 * @return Collection|Part[]
 */
public function getParts(): Collection
{
    return $this->parts;
}

public function addPart(Part $part): self
{
    if (!$this->parts->contains($part)) {
        $this->parts[] = $part;
    }

    return $this;
}

public function removePart(Part $part): self
{
    $this->parts->removeElement($part);

    return $this;
}
```

```php
/**
 * @return Collection|Sale[]
 */
public function getSales(): Collection
{
    return $this->sales;
}

public function addSale(Sale $sale): self
{
    if (!$this->sales->contains($sale)) {
        $this->sales[] = $sale;
        $sale->setCar($this);
    }

    return $this;
}

public function removeSale(Sale $sale): self
{
    if ($this->sales->removeElement($sale)) {
        // set the owning side to null (unless already changed)
        if ($sale->getCar() === $this) {
            $sale->setCar( car: null);
        }
    }

    return $this;
}
```

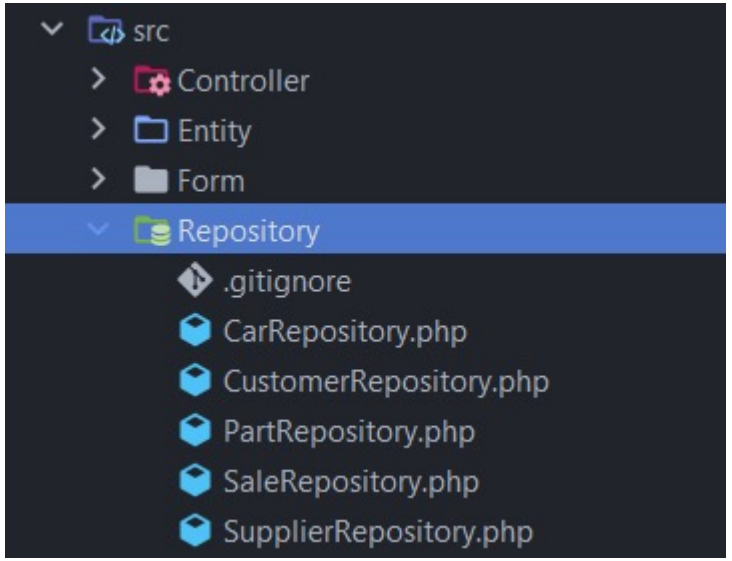
ENTITY REPOSITORY

Entity Repository, Helper Methods

UNIVERSITY of GREENWICH
Alliance with FPT Education

Entity Repository

- Makes your code reusable
- Isolates your queries from other logic
- Provides access to helper methods for each property of your entity
- Where the EntityRepository class lives?



Entity Repository Class

- How to create?
  - Automatically created when executing:

```
php bin/console make:entity SomeClass
```

Command generates Doctrine Entity and empty repository class

Entity Repository Class

- How to create?
  - Manually configure repositoryClass in your entity

```php
<?php

namespace App\Entity;

use ...

/**
 * @ORM\Entity(repositoryClass=CarRepository::class)
 */
class Car
```

```php
class CarRepository extends ServiceEntityRepository
```

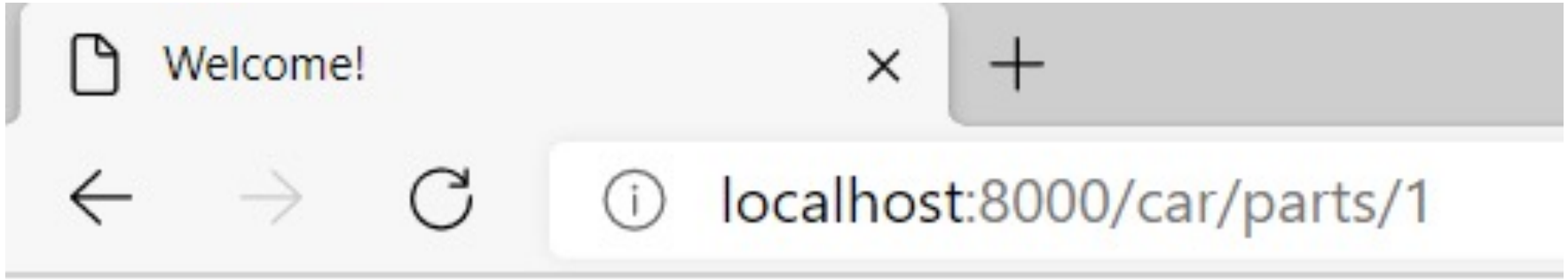
UNIVERSITY of GREENWICH
Alliance with FPT Education

Entity Repository Class

- Extending Doctrine\ORM\EntityRepository allows us to access dynamic helper methods for each mapped entity property
- Now you have dynamic method names for our entity properties without writing single line of code

```php
/**
 * @Route("/car/parts/{id}", name="car_parts")
 */
public function getParts($id){

    $em = $this->getDoctrine()->getManager();

    $repo = $em->getRepository( className: Car::class);

    $result = $repo->find( id: 1)->getParts();

    return $this->render( view: 'car/part.html.twig',[
        'data' => $result
    ]);
}
```

Entity Repository Class



- Valance
- Center-locking
- Engine shake damper and vibration absorber
- Radiator (fan) shroud
- Exhaust pipe

Entity Repository Class

- How to create?
  - Create repository class in src/Repository/Repository

```php
<?php

namespace App\Repository;

use ...

/** @method Car|null find($id, $lockMode = null, $lockVersion = null) ... */
class CarRepository extends ServiceEntityRepository
{
    public function __construct(ManagerRegistry $registry)
    {
        parent::__construct($registry, entityClass: Car::class);
    }

    /**
     * @return Car[] Returns an array of Car objects
     */
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Entity Repository Class

- Querying with the Query Builder

```
function getCarsByMake($make){
  return $this→createQueryBuilder( alias: 'car')
    →where( predicates: 'car.make = :make')
    →setParameter( key: 'make', $make)
    →getQuery()
    →getResult();
}
```

```
/**
 * @Route("/car/findbymake/{name}",name="car_find_by_make")
 */
public function findByMake($name)
{
  $em = $this→getDoctrine()→getManager();

  $repo = $em→getRepository( className: Car::class);

  $data = $repo→getCarsByMake($name);
  return $this→render( view: 'car/index.html.twig', array(
    'cars' ⇒ $data,
  ));
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

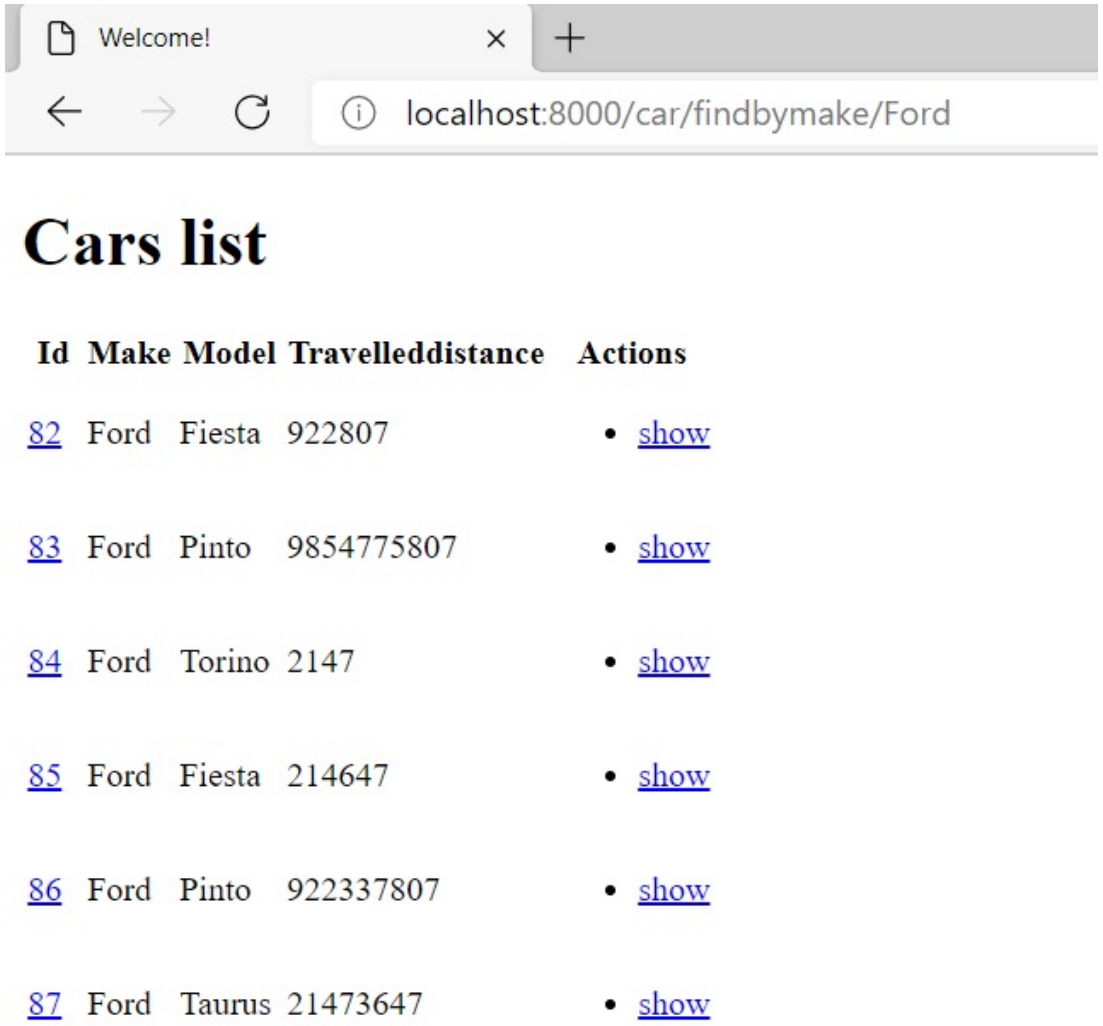
Querying with SQL

- You can query directly with SQL if you need to:

```php
// src/Repository/ProductRepository.php

// ...
class ProductRepository extends ServiceEntityRepository
{
    public function findAllGreaterThanPrice(int $price): array
    {
        $conn = $this->getEntityManager()->getConnection();

        $sql = '
            SELECT * FROM product p
            WHERE p.price > :price
            ORDER BY p.price ASC
            ';
        $stmt = $conn->prepare($sql);
        $stmt->execute(['price' => $price]);

        // returns an array of arrays (i.e. a raw data set)
        return $stmt->fetchAllAssociative();
    }
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Entity Repository Class



UNIVERSITY of GREENWICH
Alliance with FPT Education

Summary

- Doctrine ORM
- Basic entity mapping
- DBAL
- Entity Manager
- QueryBuilder
- DQL
- Entity Repository