

فهاد ال projet الجديد لي سبق ودرناه فالقناة غادي نشوفوا كيفاش نقادو blog ب 4 symfony ولي غادي نشوفوا فيه تقريبا كل ما يخص 4 symfony بالدارجة.  
فهاد الجزء الأول غادي نشوفوا كيفاش نقادو قاعدة البيانات ونربطوا بها الإتصال منبع غادي نزيدو أول table ديالنا.

## 1- إضافة قاعدة البيانات

---

كيف قلنا قبل ل projet سبق وزدناه فالقناة ممكن تشوفوا من [هنا](#) باش يلا كان شي نقص فالشرح هنا تلقاه تما.

إذن كيف قلنا غادي نزيدو قاعدة البيانات أول حاجة غادي تفتح cmd فل projet ديالك وتدير هاد les commandes :

```
composer require symfony/orm-pack
composer require --dev symfony/maker-bundle
```

منبع غادي تمشي للملف .env. تما فين عندي la connexion مع قاعدة البيانات غادي تزيد إسم قاعدة البيانات انا سميتها sym-blog ويلا كنتي داير كلمة المرور باش كتكونيكتا مع phpmyadmin ديها بلاصت '!'.

التعديلات لي غادي تزيد ف .env :

```
//.env file
//update this line
DATABASE_URL=mysql://root:''@127.0.0.1:3306/sym-blog
```

## 2- إضافة post table

---

من بعد مدرنا الإتصال مع قاعدة البيانات فحنا مازال مزدناهاش باش تزيدها غادي تدير هاد la : commande

```
php bin/console doctrine:database:create
```

دبا غادي نزيديو أول Entity لي غادي نسميها Post فغادي تدير هاد ل commande :

```
php bin/console make:migration
```

منبعد غادي تمشي ل dossier Entity غادي تلقى الملف Post.php تزد فيه غادي يكونوا عندي الحقول لي غادي تاخدمهم table post.

وعندي أيضا العلاقات مع ل user & comments بالنسبة ل likedBy غادي نشوفوها من بعد.

الكود ديال الملف Post.php هو :

```
//Post.php
```

```
<?php
```

```
namespace App\Entity;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass="App\Repository\PostRepository")
 * @ORM\HasLifecycleCallbacks()
 */
class Post
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

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

```

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

/**
 * @ORM\Column(type="datetime")
 */
private $time;

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

/**
 * @ORM\ManyToOne(targetEntity="App\Entity\User", inversedBy="posts")
 * @ORM\JoinColumn()
 */
private $user;

/**
 * @ORM\OneToMany(targetEntity="App\Entity\Comment", mappedBy="post")
 * @ORM\OrderBy({"created" = "DESC"})
 */
private $comments;

/**
 *
 * @ORM\ManyToMany(targetEntity="App\Entity\User", inversedBy="postsLiked")
 * @ORM\JoinTable(name="post_likes",
 *     joinColumns={@ORM\JoinColumn(name="post_id",
referencedColumnName="id")},
 *     inverseJoinColumns={@ORM\JoinColumn(name="user_id",
referencedColumnName="id")})
 * )
 */
private $likedBy;

public function __construct(){
    $this->likedBy = new ArrayCollection();
    $this->comments = new ArrayCollection();
}

/**
 * @return Collection
 */
public function getLikedBy(){

```

```
        return $this->likedBy;
    }

    /**
     *@return Collection
     */
    public function getComments(){
        return $this->comments;
    }

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

    public function getTitle(): ?string
    {
        return $this->title;
    }

    public function setTitle(string $title): self
    {
        $this->title = $title;

        return $this;
    }

    public function getBody(): ?string
    {
        return $this->body;
    }

    public function setBody(string $body): self
    {
        $this->body = $body;

        return $this;
    }

    public function getTime(): ?\DateTimeInterface
    {
        return $this->time;
    }

    public function setTime(\DateTimeInterface $time): self
    {
        $this->time = $time;

        return $this;
    }
}
```

```

public function getSlug(): ?string
{
    return $this->slug;
}

public function setSlug(string $slug): self
{
    $this->slug = $slug;

    return $this;
}

public function getUser()
{
    return $this->user;
}

public function setUser($user): self
{
    $this->user = $user;

    return $this;
}
//set time evry time we add a record
/**
 * @ORM\PrePersist
 */
public function setTimeValue()
{
    $this->time = new \DateTime();
}
//check if current user already like a post
public function like(User $user){
    if($this->getLikedBy()->contains($user)){
        return false;
    }
    $this->getLikedBy()->add($user);
}
}

```

**3- إضافة ل migration**

فباش نزيد ل Entity فقاعدة البيانات خصني نزيد migration فغادي ندير هاد ل commande :

**php bin/console make:migration**

منبع غادي تمشي ل dossier Migrations غادي تلقها تزدت تما فيها غادي تلقها هاد الكود :

```
//
<?php

declare(strict_types=1);

namespace DoctrineMigrations;

use Doctrine\DBAL\Schema\Schema;
use Doctrine\Migrations\AbstractMigration;

/**
 * Auto-generated Migration: Please modify to your needs!
 */
final class Version20190311122335 extends AbstractMigration
{
    public function getDescription() : string
    {
        return '';
    }

    public function up(Schema $schema) : void
    {
        // this up() migration is auto-generated, please modify it to your
needs
        $this->abortIf($this->connection->getDatabasePlatform()->getName()
!=='mysql', 'Migration can only be executed safely on \'mysql\'.');

        $this->addSql('CREATE TABLE post (id INT AUTO_INCREMENT NOT NULL,
title VARCHAR(255) NOT NULL, body LONGTEXT NOT NULL, slug VARCHAR(255) NOT
NULL, time DATETIME NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci ENGINE = InnoDB');
    }

    public function down(Schema $schema) : void
    {
        // this down() migration is auto-generated, please modify it to your
needs
        $this->abortIf($this->connection->getDatabasePlatform()->getName()
!=='mysql', 'Migration can only be executed safely on \'mysql\'.');

        $this->addSql('DROP TABLE post');
    }
}
```

4- إضافة des posts عشوائيين فقاعدة البيانات

---

كيف شفنا زدنا ل migration باش نرسلها وتزاد ك table فقاعدة البيانات غادي ندير هاد ل  
: commande

### **php bin/console doctrine:migrations:migrate**

منبعد غادي تمشي لقاعدة البيانات غادي تلقى post table تزاد.

دبا غادي نزيدو معلومات عشوائية ف la table post باش ندير هادشي غادي نخدم بل fixtures.  
أول حاجة خصني ن installer ل package فكندير هاد ل commande :

### **composer require --dev doctrine/doctrine-fixtures-bundle**

منبعد غادي نزيد أول fixture سميتها PostFixture غادي تدير هاد ل commande :

### **php bin/console make:fixtures**

منبعد غادي تمشي ل dossier DataFixtures غادي تلقاها تزاد تما فيها غادي تلقا هاد الكود :

```
//
<?php

namespace App\DataFixtures;
use App\Entity\Post;
use Doctrine\Bundle\FixturesBundle\Fixture;
use Doctrine\Common\Persistence\ObjectManager;
use Cocur\Slugify\Slugify;

class PostFixture extends Fixture
{
    public function load(ObjectManager $manager)
    {
        $slugify = new Slugify();
        for($i=0;$i<10;$i++){
            $post = new Post();
            $post->setTitle('Lorem ipsum post number '.rand(0,100));
            $post->setBody('Lorem ipsum dolor sit amet consectetur
adipisicing elit. Aperiarn earum et quasi illo quidem dicta, officia ipsam
molestiae nesciunt nemo vero voluptatibus autem necessitatibus odit expedita
exercitationem sunt, nostrum magnam? '.rand(0,100));
            $post->setTime(new \DateTime());
            $post->setSlug($slugify->slugify($post->getTitle()));
        }
    }
}
```

```
        $manager->persist($post);
    }
    $manager->flush();
}
}
```

## 5- إضافة des posts عشوائيين فقاعدة البيانات تتمة

---

كيف شفنا زدنا fixture لي غادي تمكني من إضافة des posts ولي هما 10 فقاعدة البيانات ديالي . هنا خدمت ب package سميتو [cocur/slugify](#) ولي كيمني باش نحول العنوان ديال ل article ل slug.

باش تزيدو غادي تدير هاد ل commande :

**composer require cocur/slugify**

ولي هو لي خدمنا به فل PostFixture باش حولنا title ل slug.

أخيرا باش ترسل ل fixture ل قاعدة البيانات وتزيد les articles ف table post غادي تدير هاد ل commande :

```
php bin/console doctrine:fixtures:load //
```