

# Support de cours : Les Fixtures en Symfony

## 1. Introduction aux Fixtures

Les **fixtures** permettent d'insérer des données de test dans la base de données lors du développement d'une application Symfony. Cela facilite le remplissage de la base avec des jeux de données réalistes pour tester les fonctionnalités.

Symfony utilise le package **doctrine/doctrine-fixtures-bundle** pour gérer les fixtures.

### Installation du bundle :

Si ce n'est pas déjà fait, commencez par installer le bundle :

```
composer require --dev doctrine/doctrine-fixtures-bundle
```

Activez le bundle si nécessaire :

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

## 2. Création des Entités

Voici trois entités : **User**, **Category**, et **Conference**.

### a. Entité User

```
namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity]
class User
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 180, unique: true)]
    private ?string $email = null;

    #[ORM\Column]
    private array $roles = [];

    #[ORM\Column]
    private ?string $password = null;

    // Getters et setters
}
```

### b. Entité Category

```
namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity]
class Category
{
}
```

```

#[ORM\Id]
#[ORM\GeneratedValue]
#[ORM\Column]
private ?int $id = null;

#[ORM\Column(length: 255)]
private ?string $name = null;

// Getters et setters
}

```

### c. Entité Conference

```

namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity]
class Conference
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 255)]
    private ?string $title = null;

    #[ORM\ManyToOne(targetEntity: Category::class)]
    private ?Category $category = null;

    #[ORM\ManyToOne(targetEntity: User::class)]
    private ?User $organizer = null;

    // Getters et setters
}

```

## 3. Création des Fixtures

### a. Configuration de la classe de Fixtures

Créer une classe AppFixtures dans le dossier src/DataFixtures :

```

namespace App\DataFixtures;

use App\Entity\User;
use App\Entity\Category;
use App\Entity\Conference;
use Doctrine\Bundle\FixturesBundle\Fixture;
use Doctrine\Persistence\ObjectManager;
use Symfony\Component\PasswordHasher\Hasher\UserPasswordHasherInterface;

class AppFixtures extends Fixture
{
    public function __construct(private UserPasswordHasherInterface
$passwordHasher) {}

    public function load(ObjectManager $manager): void
    {
        // Création des catégories
        $categories = [];
        foreach (['Symfony', 'PHP', 'Docker'] as $catName) {
            $category = new Category();

```

```

        $category->setName($catName);
        $manager->persist($category);
        $categories[] = $category;
    }

    // Création des utilisateurs
    $user = new User();
    $user->setEmail('organisateur@example.com');
    $user->setRoles(['ROLE_ADMIN']);
    $user->setPassword($this->passwordHasher->hashPassword($user,
'password'));
    $manager->persist($user);

    // Création des conférences
    for ($i = 1; $i <= 5; $i++) {
        $conference = new Conference();
        $conference->setTitle("Conférence $i");
        $conference->setCategory($categories[array_rand($categories)]);
        $conference->setOrganizer($user);
        $manager->persist($conference);
    }

    $manager->flush();
}
}

```

## b. Charger les fixtures

Exécutez la commande suivante pour insérer les données :

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

❑ **Attention :** Cette commande vide la base de données avant d'insérer les nouvelles données.

Si vous voulez éviter la confirmation interactive :

```
bin/console doctrine:fixtures:load --no-interaction
```

## 4. Structurer plusieurs classes de Fixtures

Si vos fixtures deviennent volumineuses, vous pouvez les diviser en plusieurs classes en utilisant `Fixture` et `DependentFixtureInterface`.

Exemple :

```

namespace App\DataFixtures;

use Doctrine\Bundle\FixturesBundle\Fixture;
use Doctrine\Common\DataFixtures\DependentFixtureInterface;
use Doctrine\Persistence\ObjectManager;

class ConferenceFixtures extends Fixture implements DependentFixtureInterface
{
    public function load(ObjectManager $manager): void
    {
        // Création de conférences liées aux catégories existantes
    }

    public function getDependencies(): array
    {
        return [CategoryFixtures::class];
    }
}

```

```
}  
}
```

## 5. Conclusion

Les **fixtures** sont essentielles pour alimenter votre base de données avec des données de test et valider vos fonctionnalités en environnement de développement. En les structurant correctement, vous facilitez la gestion et l'évolution de votre application Symfony.