

## B2.1-> TP3 Symfony Modèles, Entités et ORM

1. Créez l'entité Etudiant avec les caractéristiques suivantes : nom, prénom, naissance et niveau.

```
See 'docker --help'
sio@mint-vm:~/symfony-docker$ docker exec -it symfony-docker-php-1 /bin/bash
root@33df0e312c3c:/app# symfony console make:entity

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

created: src/Entity/Etudiant.php
created: src/Repository/EtudiantRepository.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):
> nom

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

Field length [255]:
> prenom

[ERROR] Invalid length "prenom".

Field length [255]:
> 255

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

updated: src/Entity/Etudiant.php

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

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/Etudiant.php
```

```

updated: src/Entity/Etudiant.php

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

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/Etudiant.php

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

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/Etudiant.php

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

```

```

root@33df0e312c3c:/app# symfony console make:migration
ParseError {#310
  #message: "syntax error, unexpected token "public""
  #code: 0
  #file: "./src/Controller/EtudiantController.php"
  #line: 39
  trace: {
    ./src/Controller/EtudiantController.php:39 {
      Composer\Autoload\{closure}^
      > #[Route('/etudiant/{id}', name: 'etudiant_id')]
      > public function afficher(ManagerRegistry $doctrine, int $id): Response
      >
    }
  }
}

```

2. Créer un nouvel attribut mail long de 100 caractères, éventuellement null.

```

#[ORM\Column(length:100)]
private ?string $mail = null;

```

Colonne	Type	Commentaire
<b>id</b>	integer	
<b>nom</b>	character varying(255)	
<b>prenom</b>	character varying(255)	
<b>naissance</b>	date	
<b>niveau</b>	integer	
<b>return</b>	character varying(255)	
<b>mail</b>	character varying(100)	

3. Faut ajouter à l'intérieur du contrôleur un nouvel objet Etudiant et vérifier la création du nouvel étudiant directement dans la console.

```

<?php

namespace App\Controller;

use App\Entity\Etudiant;
use DateTime;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Doctrine\Persistence\ManagerRegistry;

class EtudiantController extends AbstractController
{
    #[Route('/etudiant/creer/', name: 'etudiant_creer')]
    public function createEtudiant(ManagerRegistry $doctrine): Response
    {
        // Récupère l'EntityManager de Doctrine, nécessaire pour interagir avec la base de données
        $entityManager = $doctrine->getManager();

        // Création de l'étudiant
        $etudiant = new Etudiant();
        $etudiant->setNom('JOHN');
        $etudiant->setPrenom('Lee');
        $etudiant->setNaissance(new DateTime("2004-09-13"));
        $etudiant->setNiveau(1);
        $etudiant->setReturn(1);
        $etudiant->setMail('bacarhoumadi031@gmail.com');

        // Prépare Doctrine à sauvegarder l'étudiant (mais n'exécute pas encore la requête SQL)
        $entityManager->persist($etudiant);

        // Exécute la requête SQL pour insérer les données dans la base de données
        $entityManager->flush();

        return new Response('Nouvel étudiant sauvegardé, ID '.$etudiant->getId());
    }
}

```

```

root@33df0e312c3c:/app# symfony console dbal:run-sql 'SELECT * FROM etudiant'

```

```

[OK] The query yielded an empty result set.

```

#### 4. Nouvel étudiant

Nouvel étudiant sauvegardé, ID 22

#### 5. Afficher la base de donner

SELECT \* FROM "etudiant" LIMIT 50 (0.005 s) Modifier

<input type="checkbox"/> Modification	id	nom	prenom	naissance	niveau	return	mail
<input type="checkbox"/> modifier	15	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	16	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	17	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	18	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	19	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	20	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	21	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	22	JOHN	Lee	2004-09-13	1	bacar@gmail.com	bacar@gmail.com
<input type="checkbox"/> modifier	23	JOHN	Lee	2004-09-13	1	1	bacarhoumadi031@gmail.com
<input type="checkbox"/> modifier	24	JOHN	Lee	2004-09-13	1	1	bacarhoumadi031@gmail.com
<input type="checkbox"/> modifier	25	JOHN	Lee	2004-09-13	1		bacarhoumadi031@gmail.com











6. Affiche la modification

Modification du niveau pour 32 étudiant(s).

```
// Modifications multiples :  
[Route('/changement', name: 'changement_niveau')]  
public function changementNiveau(  
    EntityManagerInterface $entityManager,  
    EtudiantRepository $etudiantRepository): Response
```

## Tâches

 Par projet  Tableau kanban  Toutes les tâches

Aa Nom de la tâche	🌟 État
 Rédiger une proposition de pr	● Terminé
 Planifier la réunion de lancem	● En cours
 Méthode SCRUM	● En cours
 Contexte	● En cours
 Installation Docker	● En cours
 Liste des propriétaires	● Pas comme...
 Fiche Voiture	● Pas comme...
 Attribution des rôles	● Terminé
 Création de la facture	● Pas comme...
 Création de la VM	● Terminé

Aider mon binôme :

- Vous souhaitez travailler avec votre équipe ? Utilisez le menu « Partager » en haut à droite.
- Utilisez le champ « Personne assignée » ci-dessus pour attribuer des tâches à vos coéquipiers. Les personnes concernées verront une notification s'afficher dans leur barre latérale.