

Lancer un projet Symfony

Dans le terminal, commande :

1) architecture de base pour un projet website =>

```
composer create-project symfony/website-skeleton super_projet
```

Avec cette commande composer installe la dernière version de symfony, pour installer une autre version mettre la version précédée d'un ^ :

```
composer create-project symfony/website-skeleton my_project ^5.2.*
```

ou avec le officiel Symfony CLI =>

```
symfony new --full mon-super-projet
```

2) Démarrer le server =>

```
symfony server:start  
ou symfony serve
```

3) Configurer la database_url dans le fichier .env :

```
22 # MAILER_DSN=smtplib://localhost  
23 ###< symfony/mailer ###  
24  
25 ###> doctrine/doctrine-bundle ###  
26 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html  
27 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml  
28 #  
29 # DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"  
30 DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7"  
31 # DATABASE_URL="postgresql://symfony:ChangeMe@127.0.0.1:5432/app?serverVersion=13&charset=utf8"  
32 ###< doctrine/doctrine-bundle ###  
33
```

Mettre root à la place de db_user et supprimer le password si il n'y a pas de password, donner un nom à la bdd, mettre la bonne version du server.

4) Créer une base de donnée => Le nom de la bdd sera celui qui a été mis dans DATABASE_URL.

```
doctrine:database:create
```

5) Mettre dans le dossier public le dossier assets et cetera

6) Dans le dossier config/package/twig.yaml ajouter la ligne

```
form_themes: ['bootstrap_4_layout.html.twig']
```

dans le cas d'utilisation de bootstrap.

Il est nécessaire d'ajouter ce thème parce que les formulaires créés avec les commandes ne prennent pas en compte les classes ajoutées manuellement dans les vues.

7) Créer une page Home =>

```
symfony console make:controller home
```

7bis) Dans HomeController mettre @Route("/", name="home"), de manière à ce que la page d'accueil du serveur soit le page Home

Créer des articles avec un form affichés dans une vue article

1) Créer une entity (table(s) pour la bdd + colonne(s)) =>

```
symfony console make:entity superEntity
    nom
    type
    taille
    null ou pas
```

La classe **SuperEntity** est créée dans **src/Entity/SuperentityType** avec les colonnes correspondantes ainsi que **SuperEntityRepository** dans **src/Repository/SuperEntityRepository**

Entity et Repository correspondent au Modèle dans l'architecture MVC ils sont en lien direct avec la bdd

2) Push la SuperEntity dans la bdd =>

```
symfony console doctrine:schema:update --force
```

met à jour la bdd apres avoir créé les entity, pas d'historique

3) Créer un **ArticleController** =>

```
symfony console make:controller article
```

Le controller et la vue dans **templates/article/index.html.twig** sont créés.

4) Rendre possible l'ajout d'un article dans la bdd à partir d'un formulaire avec =>

```
symfony console make:form article(nom de la class du form) Article(de l'entité )
```

5) On crée la vue dans **Templates/article/addArticle.html.twig**

6) On code la méthode addArticle dans la class ArticleController :

```
/**
 * @Route("/ajouter-article", name="addArticle" )
```

```

*/
public function addArticle(Request $request , EntityManagerInterface $manager): Response
{
    // je crée un objet article
    $article= new Article();
    // je crée le formulaire du type ArticleType lié a l'objet article
    $form= $this->createForm(ArticleType::class,$article);
    // je le met a l'ecoute de la request
    $form->handleRequest($request);
    // je test si le formuliare a était submit
    if($form->isSubmitted()){
        // je test si le formulaire est valide
        if($form->isValid()){

            // j'envoie a la base de donnée (insert into )
            $manager->persist($article);
            $manager->flush();
            // une fois inseré je retourne vers la page article
            return $this->redirectToRoute('article');
        }
    }

    return $this->render('article/addArticle.html.twig',
    [
        // j'envoi le formulaire a la vue
        'form' => $form->createView()
    ]);
}

```

Authentification, connexion, déconnexion

1) Pour créer la table user dans la BDD =>

```

symfony console make:user
    user
    yes
    email
    yes

```

Pour ajouter des colonnes (noms, prénom, adresse...) faire un

```

symfony console make:entity user

```

2)Push dans la BDD =>

```

symfony console doctrine:schema:update --force

```

3)Créer l'authentification

```
symfony console make:auth
1
AppCustomAuthenticator
SecurityController
yes
```

4) dans `src/security/AppCustomAuthenticator.php` aller dans la méthode `onAuthenticationSuccess` et y modifier la ligne en mettant :

```
return new RedirectResponse($this->urlGenerator->generate('home')); ----- changer
"some_route" par "home"
```

Puis aller dans `config/packages/security.yaml`
ligne 27 => `target: home`

5) Créer un formulaire d'inscription =>

```
symfony console make:registration-form
yes
no
yes
```

La **formulaire est créé** dans `src/Form/RegistrationFormType.php` et dans la vue `templates/registration/register.html.twig`

dans Form/ `RegistrationFormType.php` ajouter les colonnes de l'entity User éventuellement ajoutées en 1)

```
avec ->add(child:'nom', type:TextType::class)
->add(child:'prenom', type:TextType::class)
```