

## Compte-rendu du test technique

Pour répondre aux attentes de l'énoncé j'ai utilisé Symfony 5, Doctrine, PHP 8 pour développer l'application, MySQL 8.0.26 comme système de gestion de base données, le tout sur un environnement Linux/Ubuntu. Concernant le rendu visuel, j'ai utilisé Twig comme moteur de template ainsi que Bootstrap 5 et du CSS pour le design. De plus, j'ai utilisé Git pour la gestion du versioning ainsi Composer comme gestionnaire de dépendances. J'ai hésité à utiliser un bundle de type EasyAdmin, mais j'ai fait le choix de tout coder "from scratch" pour pouvoir m'entraîner et prendre en compte les nouveautés proposées par Symfony 5 et PHP 8.

Après avoir créé la base de données avec ses différentes entités et leurs relations, j'ai mis en place les différentes routes dans les controllers. J'ai ensuite mis en place des fixtures afin de peupler la base de données avec des données factices pour pouvoir effectuer des tests. J'ai également créé un module de connexion/inscription, et mis en place des formulaires d'ajout/modification de certaines données. Enfin j'ai mis en place un système de rôle permettant d'autoriser l'accès à certaines pages et fonctionnalités en fonction de si l'utilisateur est connecté ou non et si il a un rôle Admin. J'ai également mis en place des contraintes de validations sur les formulaires. Enfin, j'ai entré des données manuellement et procédé à des tests pour vérifier le bon fonctionnement de l'ajout/modification/suppression des données.

### Liste des routes :

Nom de la route	Méthode	Path	Fonction
home	GET	/	Page d'accueil
app_login		/login	Connexion
app_logout		/logout	Déconnexion
register	POST	/register	Inscription
team_browse	GET	/team	Affiche la liste des équipes
team_read	GET	/team/{id}	Affiche les données d'une équipe
team_add	POST	/team/add	Ajout d'une équipe
team_edit	PUT	/team_edit/{id}	Edition d'une équipe
team_delete	DELETE	/team/delete/{id}	Suppression d'une équipe
player_browse	GET	/player	Affiche la liste des joueurs
player_read	GET	/player/{id}	Affiche la fiche d'un joueur

player_add	POST	/player/add	Ajout d'un joueur
player_edit	PUT	/player/edit/{id}	Edition d'un joueur
player_delete	DELETE	/player/delete/{id}	Suppression d'un joueur
matchplayer_add	POST	/matchplayer/add	Ajout des statistiques d'un joueur
matchplayer_edit	PUT	/matchplayer/edit/{id}	Edition des statistiques d'un joueur
matchplayer_delete	DELETE	/matchplayer/delete/{id}	Suppression des statistiques d'un joueur

### **Points d'amélioration :**

- Dans le formulaire d'ajout des statistiques, au lieu de devoir choisir l'id du joueur, il faudrait que les statistiques soient automatiquement liées au joueur en question lorsque l'on clique sur "ajouter des statistiques"
- je n'ai pas mis en place la possibilité de pouvoir changer la saison courante permettant d'afficher uniquement les joueurs inscrits dans l'effectif d'une équipe sur une saison donnée.
- revoir le type pour birthdate dans l'entité Player. Je l'avais laissé en chaîne de caractère dans un premier temps mais il faudrait le passer en Date.
- A noter, si le mécanisme de hachage du mot de passe est déjà en place, il serait utile de mettre en place des contraintes supplémentaires sur le mot de passe (Au moins 8 caractères dont 1 chiffre, une majuscule, une minuscule et un caractère spécial par exemple)