

---

# Génie logiciel - Projet 2022

Création d'une application web intégrant une API

---

## Application de recensement des records du monde



## Table de matières

<b>Introduction</b>	<b>3</b>
<b>1. Procédure d'installation et de test</b>	<b>3</b>
<b>2. Notre application : Sports et records</b>	<b>4</b>
<b>3. Modélisation des données</b>	<b>4</b>
a. Diagramme des classes	5
b. Modèle relationnel	5
<b>4. Documentation de l'API</b>	<b>6</b>
a. API renvoyant la liste de tous les athlètes	6
b. API renvoyant les informations sur un athlète	6
c. API pour l'ajout d'athlète	7
d. API pour la suppression d'athlète	7
e. API pour la modification d'athlète	7
<b>5. Gestion de projet</b>	<b>8</b>
a. Planning et répartition des tâches	8
b. Difficultés rencontrées	8
<b>Bilan et perspectives sur le projet</b>	<b>9</b>
a. Bilan	9
b. Perspectives pour l'application mobile	9

## Introduction

Au cours du septième semestre de formation à l'Ecole Nationale Supérieure de Cognitique, nous avons participé au projet de Génie Logiciel basé sur la réalisation d'une application web intégrant des API sur le thème de notre choix. Nous avons choisi de réaliser une application qui recense les records du monde de différents athlètes dans différents sports.

### 1. Procédure d'installation et de test

Pour accéder à notre application, après avoir téléchargé et dézippé le dossier, il faut l'ouvrir dans Visual Studio Code. Ensuite, il faut ouvrir un terminal et exécuter les commandes suivantes:

```
dotnet tool install --global dotnet-ef
```

```
dotnet add package Microsoft.EntityFrameworkCore.Sqlite
```

```
dotnet add package Microsoft.EntityFrameworkCore.Design
```

Enfin, exécutez la commande *dotnet watch* pour ouvrir le site. Vous pouvez à présent naviguer dans les différentes pages.

Pour exécuter les requêtes curl, il vous faudra ouvrir un deuxième terminal et y copier/coller le code présent dans le fichier *requetes curl.txt* pour ne pas avoir à les réécrire. Vous pourrez observer l'effet des requêtes directement sur le site ou en utilisant DbBrowser pour explorer la base de données.

## 2. Notre application : Sports et records

Nous avons choisi de créer une application liée au sport. Dans cette application, l'utilisateur a accès à une liste de sports, contenant chacun une liste de disciplines (voir *figure 1*). Pour chaque discipline, on retrouve les records associés (féminin et masculin), ainsi que des informations sur les athlètes qui les ont établis.

Sports et records Sports Athlètes Ajouter sport

### Bienvenue sur Sports et Records

Les meilleures performances de tous les temps !



Figure 1: Page d'accueil du site

## 3. Modélisation des données

Notre base de données est composée de 4 classes : Sport, Discipline, Athlète et Record.

- Un sport est caractérisé par son identifiant, son nom et une image. Un sport peut contenir aucune (lors de la création), une ou plusieurs disciplines.
- Les disciplines sont caractérisées par un identifiant, un nom, une brève description et elles sont associées à un sport grâce à un identifiant.
- Un athlète est caractérisé par un identifiant, son nom, son prénom, sa date de naissance, sa nationalité, son genre, et l'identifiant de la discipline dans laquelle il détient éventuellement un record. Il est possible qu'un athlète n'ait pas de record, mais il ne peut pas en détenir plusieurs car il ne peut pratiquer qu'une seule discipline.
- Un record est caractérisé par un identifiant, la date et le lieu auxquels il a été établi, l'athlète qui l'a établi, la performance réalisée, ainsi que l'identifiant de la discipline à laquelle il correspond.

### a. Diagramme des classes

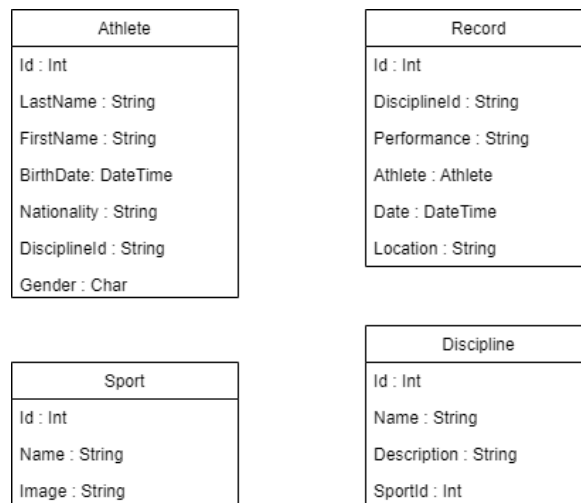


Figure 2: Diagramme de classes

### b. Modèle relationnel

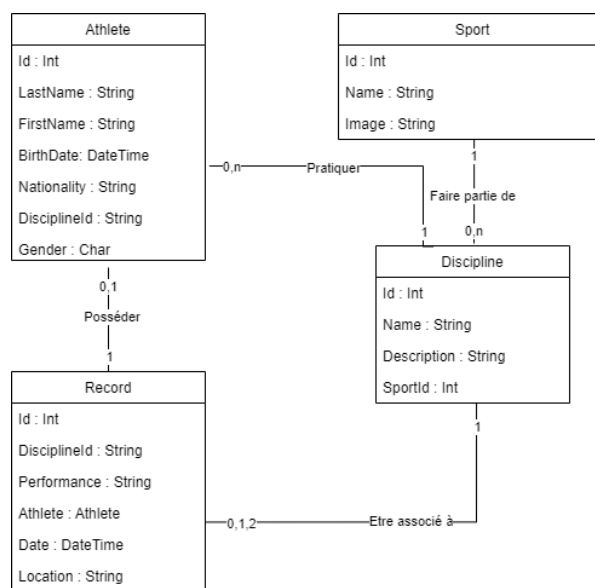


Figure 3: Modèle relationnel

## 4. Documentation de l'API

Nous avons généré une API pour le modèle Athlète : `AthleteApiController`. Nous avons alors créé 5 routes pour ce modèle.

### a. API renvoyant la liste de tous les athlètes

Le point d'accès `api/AthleteApi` renvoie les informations sur tous les athlètes sous la forme d'une liste d'objets JSON contenant les champs *id*, *lastName*, *firstName*, *gender*, *birthDate*, *nationality* et *discipline*.



```
{
  {
    id: 3,
    lastName: "Nordblad",
    firstName: "Johanna",
    gender: "F",
    birthDate: "1992-07-20T00:00:00",
    nationality: "Finlande",
    disciplineId: 0
  },
  {
    id: 7,
    lastName: "Pesquet",
    firstName: "Baptiste",
    gender: "H",
    birthDate: "1976-12-27T00:00:00",
    nationality: "France",
    disciplineId: 0
  },
  {
    id: 5,
    lastName: "Petrović",
    firstName: "Branko",
    gender: "H",
    birthDate: "1954-07-26T00:00:00",
    nationality: "Serbie",
    disciplineId: 0
  },
  {
    id: 1,
    lastName: "Powell",
    firstName: "Mike",
    gender: "H",
    birthDate: "1963-11-10T00:00:00",
    nationality: "USA",
    disciplineId: 0
  }
},
}
```

Figure 4 : Affichage écran de l'API controller de la liste des athlètes

### b. API renvoyant les informations sur un athlète

Le point d'accès `api/AthleteApi/3` renvoie les informations sur l'athlète identifié par l'id 3 sous la forme d'un objet JSON contenant les champs *id*, *lastName*, *firstName*, *gender*, *birthDate*, *nationality* et *discipline*.

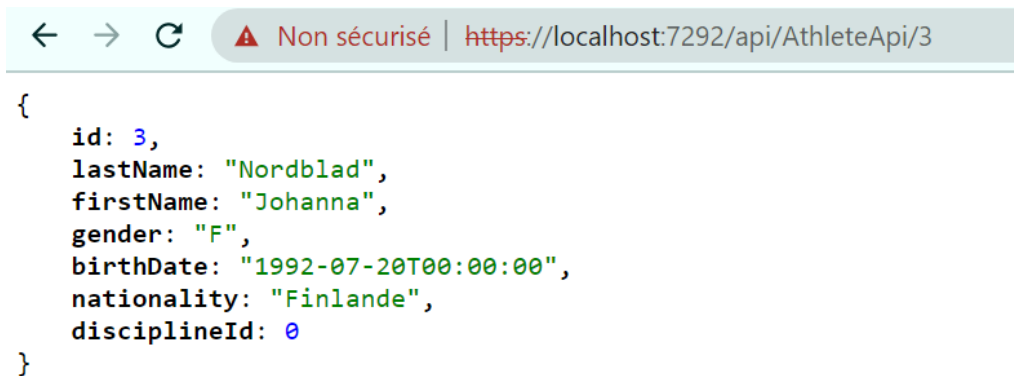


Figure 5 : Affichage écran de l'API controller de l'athlète d'id = 3

### c. API pour l'ajout d'athlète

#### Test de création avec la commande CURL :

La commande CURL suivante permet d'ajouter un athlète.

```
curl -X POST -k -H 'Content-Type: application/json' -i https://localhost:7292/api/athleteapi/
--data '{"FirstName": "Renelle", "LastName": "Lamote", "Gender": "F", "BirthDate": "1979-12-14",
"Nationality": "France", "Discipline": 2}'
```

Figure 6 : Requête CURL pour la création d'athlète

### d. API pour la suppression d'athlète

#### Test de création avec la commande CURL :

La commande CURL suivante permet de supprimer un athlète de la base de données.

```
curl -X DELETE -k -H 'Content-Type: application/json' -i https://localhost:7292/api/athleteapi/4
```

Figure 7 : Requête CURL pour la suppression d'athlète

### e. API pour la modification d'athlète

#### Test de création avec la commande CURL :

La commande CURL suivante permet de modifier un athlète dans la base de données.

```
curl -X PUT -k -H 'Content-Type: application/json' -i https://localhost:7292/api/athleteapi/1 --data
'{"Id": "1", "LastName": "Powell", "FirstName": "Mike", "Gender": "H", "BirthDate": "1963-11-10",
"Nationality": "USA", "DisciplineId": "1"}'
```

Figure 8 : Requête CURL pour la modification d'athlète

## 5. Gestion de projet

### a. Planning et répartition des tâches

Dès le début du projet, nous avons fait une liste des tâches que nous devons faire et nous nous les sommes réparties. Nous avons préféré effectuer certaines tâches en binôme afin de mieux comprendre nos erreurs et les résoudre.

Au début de ce projet, nous avons travaillé ensemble notamment lors de la phase de brainstorming et de recherche de fonctionnalités. Puis, lors des vacances, nous nous sommes répartis les tâches. Dans la mesure du possible, nous travaillons à côté afin de discuter de nos problèmes (ou suggestions) et de s'entraider.

Planning - Projet GL 2A CÔTE-COLISSON Romane RIBARD-DEL CASTILLO Célia							
Tâches \ Dates	2/12	5/12 - 11/12	12/12 - 18/12	19/12 - 25/12	26/12 - 01/01	02/01 - 08/01	
Brainstorming et choix du sujet							
Création du diagramme de classes							
Définition des fonctionnalités							En binôme
Premières maquettes des pages du site							Célia
Création du projet et du dépôt git							Romane
Création des modèles							Non réalisé
Création d'un SeedData basique							
Création de l'API athlète							
Requêtes curl							
Page Sports (contrôleur et vue)							
Page Ajout d'un sport (contrôleur et vue)							
Page Disciplines (contrôleur et vue)							
Page Ajout d'une discipline (contrôleur et vue)							
Page Records (contrôleur et vue)							
Page Ajout d'un record (avec choix de l'athlète et de la discipline existants)							
Page Athlète (contrôleur et vue)							
Page Ajout d'un athlète (avec choix d'une discipline existante)							
Agrandissement de la BDD (SeedData)							
Mise en page du site (CSS - Bootstrap)							
Rédaction du rapport							

Figure 9 : Planning du déroulement du projet avec la répartition de rôles

### b. Difficultés rencontrées

Nous avons rencontré certaines difficultés lors de ce projet. Nous avons dû nous débrouiller seules car ces problèmes arrivaient lors du développement plus avancé de l'application, lorsqu'il n'y avait plus de créneaux réservés au projet avec un professeur. Nous avons tout d'abord eu des difficultés pour récupérer des données entre les contrôleurs et les vues quand nous avons besoin d'informations de plusieurs tables pas forcément reliées entre elles. Nous avons finalement ViewData pour y parvenir. De plus, lors de l'ajout d'une discipline à la base de données, nous avons une erreur liée à l'id de la discipline (id déjà utilisé) tandis que pour l'ajout d'un sport (très ressemblant), nous n'avons pas ce problème. Les id sont pourtant censés s'auto incrémenter et nous n'avons pas compris d'où venait le problème, car les id d'une discipline et d'un sport sont définis de la même manière et apparaissent bien en auto incrément sur DbBrowser. Pour pallier ce



problème, nous générons dans le programme l'id de la discipline à créer en recherchant l'id maximal existant en base de données.

Nous avons eu l'occasion de tester l'aide de ChatGPT, une intelligence artificielle à laquelle nous pouvons poser des questions en tout genre et notamment en programmation. Cette aide nous a été bien utile, bien qu'il faille parfois préciser les questions pour obtenir une réponse satisfaisante.

## Bilan et perspectives sur le projet

### a. Bilan

La réalisation du back-end à l'aide de la technologie ASP.NET Core MVC est une nouvelle notion que nous avons découverte lors de ce semestre. Ce projet nous a permis de mettre en pratique ces notions vues en cours et en TD en essayant de se détacher des tutoriels (et de la correction des TP). Nous avons consacré beaucoup de temps à gérer nos erreurs mais cela nous a fait apprendre et nous sommes maintenant capable d'en comprendre et d'en régler la plupart.

### b. Perspectives pour l'application mobile

Nous avons essayé de développer à la fois le back-end et l'esthétique et les fonctionnalités du site. Cependant, il reste évidemment des fonctionnalités que nous aurions aimé ajouter si nous avions eu plus de temps (et de connaissances). Par exemple, il n'est possible de créer via le site que des sports et des disciplines. Il serait bien sûr utile de pouvoir ajouter des athlètes et des records. Il serait aussi intéressant d'implémenter une fonctionnalité pour modifier et supprimer des éléments. De plus, il faudrait ajouter la possibilité pour l'utilisateur de mettre une photo lorsqu'il crée un sport. Enfin, dans le modèle actuel, un athlète ne peut exercer qu'une seule discipline. Il serait plus cohérent de faire en sorte qu'un athlète puisse exercer et avoir des records dans plusieurs disciplines d'un même sport.