



مكتب التكوين المهني و إنعاش الشغل

Office de la Formation Professionnelle  
et de Promotion du Travail

## Direction Régionale Drâa Tafilalet

### TP N°2

#### Création d'une application Cloud native

Année : 2022/2023

**Filière** : Développement Digital Option Web Full Stack

**Niveau** : Technicien Spécialisé

---

Jeudi 25 Mai 2023

#### **Exercice 1** : Création d'une API REST pour la gestion des utilisateurs.

1. Créez une application Node.js en utilisant Express.
  2. Configurez la connexion à une base de données MongoDB (**MyDB**) avec une collection (**users**) en utilisant Mongoose.
  3. Créez un modèle d'utilisateur qui contient les champs suivants: nom, email et password.
  4. Créez un endpoint **POST** pour permettre la création d'un utilisateur avec les champs suivants : nom, email et password.
  5. Créez un endpoint **GET** pour récupérer tous les utilisateurs.
  6. Créez un endpoint **GET** pour récupérer un utilisateur en utilisant son ID.
  7. Créez un endpoint **PUT** pour mettre à jour un utilisateur en utilisant son ID.
  8. Créez un endpoint **DELETE** pour supprimer un utilisateur en utilisant son ID.
-

## **Exercice 2:** Création d'une API REST pour la gestion des équipes.

Considérer les deux collections **teams** et **players**.

- **teams** -

```
[
  { "id": 1, "name": "PSG", "country": "France" },
  { "id": 2, "name": "Barcelone", "country": "Espagne" },
  { "id": 3, "name": "Real Madrid", "country": "Espagne" },
  { "id": 4, "name": "Milan", "country": "Italie" },
  { "id": 5, "name": "OM", "country": "France" }
]
```

- **players** -

```
[
  { "id": 1, "teamId": 2, "name": "Messi Lionel", "number": 10, "post": "Forward" },
  { "id": 2, "teamId": 3, "name": "Karim Benzima", "number": 9, "post": "Forward" },
  { "id": 3, "teamId": 2, "name": "Gavi Pablo", "number": 30, "post": "Midfielder" },
  { "id": 4, "teamId": 3, "name": "Valverdi", "number": 15, "post": "Midfielder" }
]
```

--> Chaque 'team' dispose des champs : id, name, country.

--> Chaque 'player' dispose des champs : id, teamId, name, number, post .

1. Développer les opérations CRUD pour l'entité 'team'.
2. Développer les opérations CRUD pour l'entité 'player'.
3. Développer la route permettant d'afficher les 'players' d'une 'team' via son id (*id de team*).
4. Développer la route permettant d'afficher l'équipe d'un joueur donné via son id (*id de player*).
5. Développer la route permettant de chercher un 'player' à partir de son name.
6. Modifier la route D (delete) de l'entité team pour qu'il soit conforme à ce qui suit : Supprimer une 'team' signifie supprimer tous les 'players' de cette 'team'.