

# TP 3 : Création d'une API de recettes de cuisine

**Durée estimée :** 3 heures

Créer une API REST pour une application de partage de recettes de cuisine en utilisant MongoDB et Express.js, avec un accent particulier sur la validation des données et les relations NoSQL.

## 1 - Configuration du projet et base de données

Créer un nouveau projet Node.js avec Express.

Configurer la connexion à MongoDB

Créer les collections suivantes :

- recipes (recettes)
- users (utilisateurs)
- categories (catégories de recettes)
- comments (commentaires)

Vous êtes libre pour la définition de la structure de vos collections. Il faut simplement que cela réponde au besoin des routes.

## 2 - Routes CRUD

Gestion des recettes

- GET /api/recipes
- GET /api/recipes/:id
- POST /api/recipes
- PUT /api/recipes/:id
- DELETE /api/recipes/:id
- GET /api/recipes/search (recherche avancée)

Fonctionnalités sociales

- POST /api/recipes/:id/like
- POST /api/recipes/:id/comment
- GET /api/users/:id/followers
- POST /api/users/:id/follow

Gestion des utilisateurs

- POST /api/users/register
- POST /api/users/login
- GET /api/users/profile
- PUT /api/users/profile

### **3 - Validation**

Utiliser express-validator pour la validation des données :

- Validation des ingrédients (quantités positives, unités valides)
- Validation des étapes de préparation
- Validation des temps de cuisson
- Validation des difficultés (facile, moyen, difficile)

### **Bonus, pistes d'amélioration**

- Intégrer une pagination à certaines de vos routes
- Ajouter un système de notation pour les recettes
- Ajouter un système de recommandation (requêtes similaires..)

## **Annexes : modèles complet proposé (en option)**

### Modèle user

```
{
  username: String,
  email: String,
  password: String,
  profilePicture: String,
  favoriteRecipes: [{ type: ObjectId, ref: 'Recipe' }],
  following: [{ type: ObjectId, ref: 'User' }],
  followers: [{ type: ObjectId, ref: 'User' }]
}
```

### Modèle Recipe

```
{
  title: String,
  description: String,
  ingredients: [{
    name: String,
    quantity: Number,
    unit: String
  }],
  steps: [String],
  cookingTime: Number,
  difficulty: String,
  servings: Number,
  author: { type: ObjectId, ref: 'User' },
  categories: [{ type: ObjectId, ref: 'Category' }],
  likes: [{ type: ObjectId, ref: 'User' }],
  comments: [{ type: ObjectId, ref: 'Comment' }],
  images: [String],
  createdAt: Date,
  updatedAt: Date
}
```