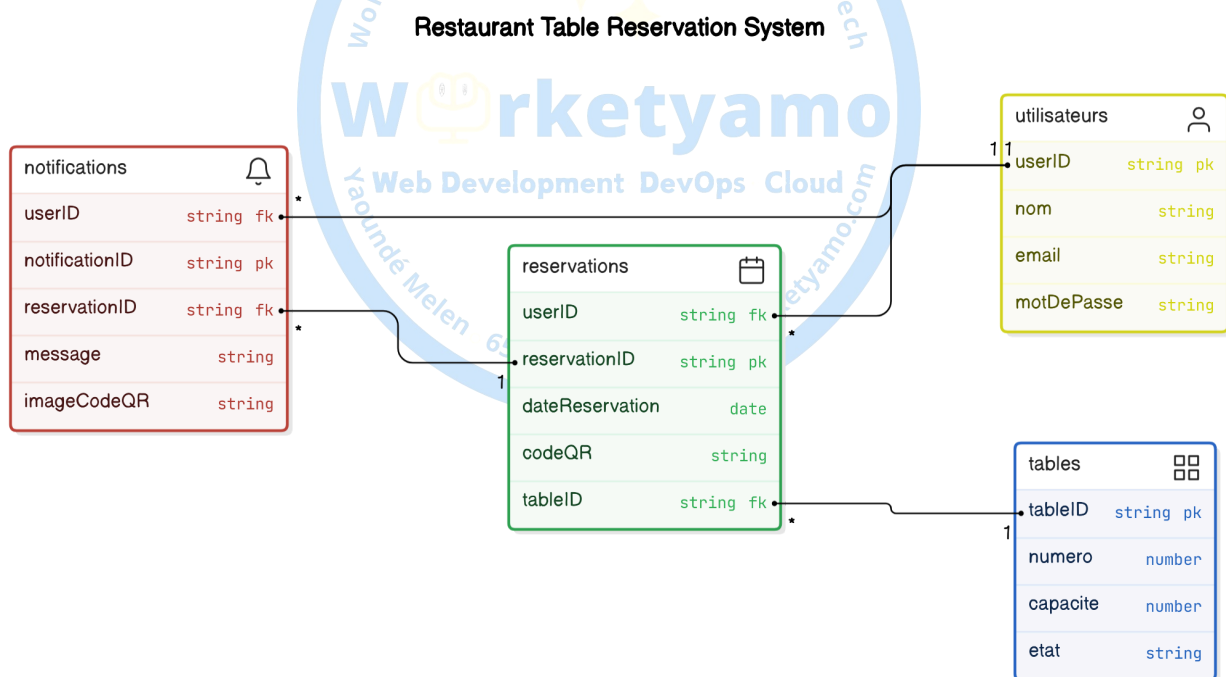


# Fiche de Travaux : Node Js - typescript

Exercice Avancé : Création d'une API REST pour la Réserveation de Tables dans un Restaurant 5 Étoiles à Yaoundé

## Diagramme



## Scénario

Vous êtes un développeur travaillant pour un restaurant 5 étoiles à Yaoundé qui souhaite permettre aux clients de réserver des tables en ligne. Chaque réservation doit générer un code QR unique, et l'état de la table doit être mis à jour et stocké dans un bucket S3. À l'entrée, le code QR est scanné, rendant la table occupée. Les utilisateurs doivent également pouvoir libérer une table, la rendant à nouveau disponible. L'API doit gérer l'authentification et envoyer des notifications par email avec l'image du code QR.

## Fonctions à Implémenter

### 1. Gestion des Utilisateurs :

- **Inscription d'un nouvel utilisateur :**
  - Endpoint : `POST /users/signup`
  - Données d'entrée : nom, email, mot de passe
  - Action : Crée un nouvel utilisateur dans la base de données, chiffre le mot de passe.
- **Connexion des utilisateurs :**
  - Endpoint : `POST /users/login`
  - Données d'entrée : email, mot de passe
  - Action : Authentifie l'utilisateur et génère un token JWT.
- **Déconnexion des utilisateurs :**
  - Endpoint : `POST /users/logout`
  - Action : Invalide le token de l'utilisateur.
- **Consulter le profil utilisateur :**
  - Endpoint : `GET /users/profile`
  - Action : Retourne les informations du profil de l'utilisateur connecté.
- **Mettre à jour le profil utilisateur :**
  - Endpoint : `PUT /users/profile`
  - Données d'entrée : nom, email, mot de passe

- Action : Met à jour les informations du profil de l'utilisateur connecté.
- **Supprimer un compte utilisateur :**
  - Endpoint : `DELETE /users/profile`
  - Action : Supprime le compte de l'utilisateur connecté.

## 2. Gestion des Tables :

- **Consulter la liste des tables disponibles :**
  - Endpoint : `GET /tables`
  - Action : Retourne la liste de toutes les tables disponibles dans le restaurant.
- **Ajouter une nouvelle table :**
  - Endpoint : `POST /tables`
  - Données d'entrée : numéro de table, capacité
  - Action : Ajoute une nouvelle table dans le restaurant.
- **Mettre à jour les informations d'une table :**
  - Endpoint : `PUT /tables/:id`
  - Données d'entrée : numéro de table, capacité
  - Action : Met à jour les informations d'une table existante.
- **Supprimer une table :**
  - Endpoint : `DELETE /tables/:id`
  - Action : Supprime une table du restaurant.

## 3. Gestion des Réservations :

- **Réserver une table :**
  - Endpoint : `POST /reservations`
  - Données d'entrée : tableID, userID, date et heure de la réservation
  - Action : Crée une réservation pour une table par un utilisateur, génère un code QR, met à jour l'état de la table à "réservée", stocke l'état de la table dans un bucket S3, et envoie une notification par email avec l'image du code QR.

- **Scanner le code QR à l'entrée :**
  - Endpoint : **POST /reservations/:id/scan**
  - Action : Rend le code QR invalide, met à jour l'état de la table à "occupée".
- **Libérer une table :**
  - Endpoint : **PUT /tables/:id/release**
  - Action : Met à jour l'état de la table à "disponible".

#### 4. Notifications :

- **Envoyer une notification avec le code QR :**
  - Action : Lorsqu'une réservation est créée, envoie une notification par email à l'utilisateur avec l'image du code QR.

#### Stack :

1. Node JS
2. Express
3. Typescript
4. Git et Github
5. MongoDB
6. Prisma ORM
7. rate-limiter
8. Nodemailer
9. qrcode
10. ejs