

# Description

Ce code regroupe les fonctions principales pour gérer les concerts dans une API Node.js connectée à une base PostgreSQL. Chaque fonction correspond à une opération CRUD (Créer, Lire, Mettre à jour, Supprimer).

La fonction `creerConcert` extrait les informations d'un concert depuis le corps de la requête HTTP, puis insère ces données dans la table `concert`. Elle initialise le nombre de places restantes avec la valeur du nombre total de places, ce qui garantit que toutes les places sont disponibles lors de la création. Si l'opération réussit, elle renvoie le concert créé avec un code HTTP 201. En cas d'erreur, elle logge l'erreur et renvoie un message d'échec.

La fonction `mettreAJourConcert` permet de modifier un concert existant identifié par son `id` (récupéré dans l'URL). Elle met à jour les champs principaux du concert. Si aucun concert n'est trouvé avec cet `id`, elle renvoie un code 404. Sinon, elle retourne les nouvelles données du concert.

La fonction `supprimerConcert` supprime un concert selon son `id`. Si le concert n'existe pas, elle renvoie un code 404. Sinon, elle confirme la suppression avec un message de succès. Les erreurs sont gérées avec un code 500.

Enfin, `obtenirConcerts` récupère tous les concerts de la base, triés par identifiant, et les renvoie au format JSON. En cas d'erreur, elle logge le problème et envoie un message d'erreur au client. Ce module centralise donc toute la logique métier liée aux concerts, ce qui facilite la maintenance et la clarté du code.

# Pseudo code des 4 fonctions du CRUD (API Node.js)

## **creerConcert**

Récupérer les données du concert depuis le corps de la requête

Essayer :

- Insérer un nouveau concert dans la base de données avec les données reçues

- Initialiser nb\_places\_restantes avec nb\_places\_total

- Renvoyer le concert créé avec un code 201

Attraper les erreurs :

- Afficher l'erreur dans la console

- Renvoyer une erreur 500 au client

## **mettreAJourConcert**

Récupérer l'id du concert depuis les paramètres de la requête

Récupérer les nouvelles données du concert depuis le corps de la requête

Essayer :

- Mettre à jour le concert correspondant dans la base de données avec les nouvelles données

- Si aucun concert n'est trouvé :

  - Renvoyer une erreur 404 au client

- Sinon :

  - Renvoyer le concert mis à jour

Attraper les erreurs :

- Renvoyer une erreur 500 au client

## **supprimerConcert**

Récupérer l'id du concert depuis les paramètres de la requête

Essayer :

- Supprimer le concert correspondant dans la base de données

- Si aucun concert n'est trouvé :

  - Renvoyer une erreur 404 au client

- Sinon :

  - Renvoyer un message de succès

Attraper les erreurs :

- Renvoyer une erreur 500 au client

## **Obtenir concert**

Essayer :

- Récupérer tous les concerts de la base de données, triés par identifiant

- Renvoyer la liste des concerts au format JSON

Attraper les erreurs :

Afficher l'erreur dans la console

Renvoyer une erreur 500

## Méthodes pour chaque fonction

### **creerConcert**

- **require()** : pour importer le module de connexion à la base.
- **query()** : méthode du pool PostgreSQL pour exécuter une requête SQL.
- **status()** : méthode de l'objet réponse Express pour définir le code HTTP.
- **json()** : méthode de l'objet réponse Express pour envoyer une réponse au format JSON.
- **console.error()** : méthode pour afficher une erreur dans la console.

### **mettreAJourConcert**

- **query()** : méthode du pool PostgreSQL pour exécuter une requête SQL.
- **status()** : méthode de l'objet réponse Express pour définir le code HTTP.
- **json()** : méthode de l'objet réponse Express pour envoyer une réponse au format JSON.

### **supprimerConcert**

- **query()** : méthode du pool PostgreSQL pour exécuter une requête SQL.
- **status()** : méthode de l'objet réponse Express pour définir le code HTTP.
- **json()** : méthode de l'objet réponse Express pour envoyer une réponse au format JSON.

### **obtenirConcerts**

- **query()** : méthode du pool PostgreSQL pour exécuter une requête SQL.
- **json()** : méthode de l'objet réponse Express pour envoyer une réponse au format JSON.
- **console.error()** : méthode pour afficher une erreur dans la console.