

# ASAC

- Frontend
- Backend - API
  - Routes
  - DB

# Frontend

# Backend - API

# Routes

- /connect
  - POST : Tente de connecter un user avec son mail/mdp et retourne un token JWT
- /users/token
  - GET : Récupère les informations du user connecté
    - nom, prénom, mail, id, sports
    - Si padel : parties, niveau, prix restant à payer
- /users
  - GET : Récupère la liste de tous les utilisateurs avec des informations limitées, peut être filtré avec des query params
    - name : nom ou partie du nom du joueur
    - level : filtre sur ce niveau
    - minlevel : affiche tous les joueurs avec au moins le niveau demandé
    - maxlevel : affiche tous les joueurs avec au plus le niveau demandé
    - sports : affiche tous les joueurs inscrits à un des sports
  - /:id
    - GET : Récupère toutes les informations publiques d'un utilisateur
- /padel
  - /subscribe
    - POST : Ajoute le padel comme sport pour le user connecté
  - /games
    - GET : Récupère la liste des parties, peut être filtré avec des query params
      - user (id) : filtre les parties auxquelles l'utilisateur spécifié à participé
      - ended (true/false) : filtre les parties terminées ou non terminées (en attente de joueurs, sans score, etc)
        - Parties terminées : score, V/D
        - Parties non terminées
          - Demandes : en attente de joueurs
          - En cours : Tous les joueurs mais pas de score
    - POST: Créé une nouvelle partie
      - double/simple
      - date et heure
      - id joueur 1
        - sélection par défaut du joueur connecté
        - utilisation de mat-autocomplete pour renseigner les joueurs (liste des utilisateurs inscrits au padel)
      - id joueur 2
      - si double
        - id joueur 3
        - id joueur 4

- /:id
  - GET : Récupère toutes les informations d'une partie
  - PUT :
    - Changement de la date ou de l'heure
    - Ajout, modification ou suppression d'un joueur
    - Fin de partie
      - Changement de la date et/ou de l'heure
      - Changement des joueurs
      - Loisir ou Match
      - Victoire, Défaite ou Match Nul pour chaque joueurs de la partie
      - Score de la partie (nombre de sets, score de chaque sets)
- /admin
  - /users
    - GET : Liste des utilisateurs étant inscrit au padel, peut être filtré avec des query params
      - Affiche pour chaque utilisateur le nombre de parties multiplié par le prix d'un ticket moins le montant déjà payé
    - /:id/montant
      - PUT : Ajoute ce montant à ce qui a été payé par l'utilisateur id
- /tickets
  - GET : Récupère la quantité de tickets disponibles
  - /add
    - POST : Ajoute un nombre de tickets au total disponible
  - /set
    - POST : Ajuste le nombre de tickets disponibles

# DB

1. users
  - id, nom, prenom, mail (unique), externe, created\_at, updated\_at, deleted\_at (nullable)
2. users\_roles
  - user\_id, role ("user", "padel-admin")
3. sports
  - id, nom (unique), created\_at, updated\_at, deleted\_at (nullable)
4. users\_sports
  - user\_id, sport\_id, created\_at, updated\_at
5. padel\_games
  - id, is\_double, game\_date, game\_hour, id\_team\_one, id\_team\_two, created\_at, updated\_at, deleted\_at (nullable)
6. padel\_games\_scores
  - id, game\_id, nb\_sets, scores (tableau json : [{"set", "team1", "team2"}]), created\_at, updated\_at, deleted\_at (nullable)
7. padel\_game\_teams
  - game\_id, user\_one\_id, user\_two\_id (nullable), side ("team1" ou "team2"), state (V, N, D, nullable), created\_at, updated\_at, deleted\_at (nullable)
  - PK : game\_id, user\_one\_id, user\_two\_id
8. padel\_game\_players
  - game\_id, user\_id, created\_at, updated\_at, deleted\_at (nullable)
  - PK : game\_id, user\_id
9. padel\_prices
  - id, ticket\_value\_cents, created\_at, updated\_at, deleted\_at (nullable, unique)
10. padel\_payments
  - id, user\_id, admin\_id, amount\_paid\_cents, ticket\_value\_cents, method ("cash", "cheque", "transer", "other"), payment\_date, created\_at, updated\_at
11. padel\_ticket\_stock
  - id, admin\_id, amount, created\_at, updated\_at