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