



Applications Web

Hagimettaceinture

Groupe L34

Élèves :

THEVENET Louis

LEBOBE Timothé

LECUYER Simon

SABLAYROLLES Guillaume

28 Juin 2025

Table des matières

1. Hagimettaceinture	3
2. Architecture	3
2.1. Strucutre	3
2.2. Entités	3
3. Blog de l'avancement	3
3.1. 21 mars 2025	3
3.2. 1 avril 2025	4
3.3. 20 mai 2025	4
3.4. 25 mai 2025	4

1. Hagimettaceinture

Le projet est une application de centralisation de circuits et d'équipes pour passionnés de courses, on peut y retrouver un ensemble de circuits renseignés par les usagers du site. Les usagers peuvent discuter au travers d'un forum de discussion et se tenir informé en s'inscrivant à une newsletter (non opérationnel). Les utilisateurs peuvent aussi communiquer les prochains événements au travers d'un calendrier et de la création de Meeting (courses, réunion, ...)

2. Architecture

2.1. Strucutre

Le projet se sépare en deux parties :

- Dossier `client` : pour les sources du front-end (réalisé en React sur un server Deno)
- Dossier `server` : le backend avec les entités et controller
- Dossier `db` : la base de données du server

Pour lancer le projet il suffit d'installer [Deno](#) et [Just](#) pour lancer le server frontend et back.

Utiliser les commandes : `just db`, `just back` et `just db` dans des terminaux différents pour lancer les différentes parties du projet.

Les différents routes du backend sont listées dans le Readme de ce projet (avec leur route et méthode)

2.2. Entités

Race `id`, *circuit*, date, *vehiculeType*, participants

Member `idMembre`, name, firstName, vehicules, subscriber

Vehicule `id`, *vehiculeType*, branch, model, licensePlate, date, *owner*

Circuit `id`, place, distance, turnNumber, spectatorNumber, name, bestTime

Meeting `id`, title, guests, date

Sponsoring `id`, *racingTeam*, *sponsor*, date, duration, investment

RacingTeam `idRacingTeam`, nom, classement, membres, sponsors

Sponsor `id`, name, investedCapital, date

Nous avons également utilisé une classe entité abstraite **Event** avec la stratégie d'héritage `InheritanceType.TABLE_PER_CLASS`.

Voici les classes qui en héritent :

Race pour sa date d'occurence,

Vehicule pour sa date de mise en service,

Circuit pour sa date de création,

Meeting pour sa date d'occurence,

Sponsoring pour ses dates de début et de fin,

Sponsor pour sa date de création.

Ceci nous permet de traiter toutes ces entités comme des événements pour les afficher sur notre page calendrier.

3. Blog de l'avancement

3.1. 21 mars 2025

- Thème de l'application : gestionnaire de sport automobile
- Création des entités

- Création des premières routes importantes (back et front)

3.2. 1 avril 2025

- UI du site web et design
- Création des premières requêtes pour récupérer les circuits et équipe
- Implémentation de l'entité **Event** pour créer différents informations sur le calendrier
- Ajout du forum et la possibilité de poster des nouveaux topics et message dans ces topics

3.3. 20 mai 2025

- Ajout de l'authentification avec JWT
- Edition des équipes et circuits
- Ajout (et création) de membres et sponsors lors de l'ajout d'une nouvelle équipe
- Liaison d'un membre avec son nom prénom lors de la création d'un compte s'il avait été ajouté à une équipe
- Dashboard d'un membre pour afficher ses informations

3.4. 25 mai 2025

- Ajout de l'autorisation :
 - Circuit : création et édition uniquement si l'utilisateur est connecté
 - Equipe : création d'une équipe si l'utilisateur est connecté, modification uniquement si le membre fait partie de l'équipe
 - Forum : ajout de topic et message si l'utilisateur est connecté
 - Calendar : ajout d'événements si l'utilisateur est connecté