

# Compte rendu de projet

## Gacha Game API

### Auteurs

- BRUNEL Bastien
- MOURONVAL Laurane
- ROSINSKI Mathys

### Outils utilisés

Nous avons travaillé avec GitHub :

<https://github.com/Mathys-R/GachaGameAPI>

### Répartition des tâches

Nous nous sommes globalement réparti les tâches de la manière suivante :

- **ROSINSKI Mathys** s'est occupé de l'API d'authentification.
- **BRUNEL Bastien** s'est occupé de l'API Monster, de l'API Summon, du front, de l'ajout automatique en BDD des monstres à l'initialisation du projet et du débogage final.
- **MOURONVAL Laurane** s'est occupée de la configuration Docker du projet, de l'API Player, du front, de l'API Summon et du débogage final.

Après avoir réfléchi à des scénarios d'exemple pour bien comprendre le projet et après avoir tout configuré afin que le projet se lance correctement, nous avons commencé à développer chacun de notre côté chaque API, puis nous avons mis en commun notre travail au fur et à mesure. Vers la fin du projet, nous avons résolu (presque) tous les problèmes.

Après relecture du sujet pour vérifier que nous ne nous étions pas éloignés des attentes, nous nous sommes rendu compte que nous avons pris une mauvaise direction pour la gestion des utilisateurs et des joueurs : nous n'avions pas lié un utilisateur à un joueur en particulier, comme demandé. Nous (Laurane et Bastien) avons donc dû refondre une partie de l'API d'authentification et de l'API Monster pour que cela fonctionne correctement.

Une autre difficulté a été de faire appel à un service d'une API depuis une autre API, ce qui était compliqué en raison de notre organisation en huit conteneurs (quatre MongoDB et quatre API). La solution trouvée a été d'effectuer des appels HTTP externes directement aux endpoints.

Précision également sur la méthode pour créer un player. La création d'un token (register/token) et celle d'un player (possédant des monstres) ne sont pas connectés (2 requêtes nécessaires). Cependant le Front rend cette opération transparente pour faciliter les tests en créant automatiquement un player lors du register d'un nouvel utilisateur. La page stocke ensuite le token et l'id du player pour faciliter les tests.

Les commentaires ont été générés par intelligence artificielle pour un gain de temps.

## **Fonctionnalités manquantes**

- Gestion des combats (optionnelle)
- Système d'XP et de niveau des monstres → Priorisation des fonctionnalités en rapport avec les parties déjà développées (pas d'API de combat)