



Bienvenue chez QiEAM Dev Back

Présentation

Nous souhaitons réaliser une API permettant de développer des applications en lien avec les jeux vidéos possédés et disponibles par les utilisateurs.

Modalités de rendu

Le résultat de votre travail doit être mis à disposition sur un système d'hébergement compatible GIT (gitlab, github, bitbucket, ...)

Envoyer un unique mail à exam@quartz-insight.com qui contient le lien d'accès à votre projet.

Le sujet du mail doit être au format suivant :

DEV BACK JAVA - NOM PRENOM

Interface

Résumé

Le service attendu doit permettre de gérer une population d'utilisateurs pouvant avoir des amis. Les utilisateurs tout comme les amis des utilisateurs possèdent des jeux. Un utilisateur peut ajouter ou supprimer un ami de sa liste d'amis. Ce service propose également un magasin en ligne de jeux vidéos que chaque utilisateur peut acheter. Les utilisateurs ne peuvent réaliser qu'une seule fois l'acquisition d'un jeu. Il se peut que certains jeux soient retirés de la boutique en ligne. Il ne faut cependant pas empêcher les utilisateurs l'ayant déjà acquis de le voir et donc d'y jouer.

Points d'accès Web API

L'application attendue est donc un service exposant une web API. Nous envisageons les points d'accès suivants

```
GET /api/store/games
POST /api/store/games
GET /api/store/games/{gameId}
DELETE /api/store/games/{gameId}
GET /api/users
POST /api/users
GET /api/users/{userId}
GET /api/users/{userId}/games
POST /api/users/{userId}/games
DELETE /api/users/{userId}/games/{gameId}
DELETE /api/users/{userId}
GET /api/users/{userId}/friends
POST /api/users/{userId}/friends
DELETE /api/users/{userId}/friends/{friendId}
```

Format d'échange

Le Content-Type attendu est application/json pour les corps éventuels de requêtes et de réponses.

Un jeu devra être représenté en JSON

```
{
  id: 0,
  title: "title",
  cover: "url"
}
```

Un utilisateur devra être représenté en JSON

```
{
  id: 0,
  username: "string",
  email: "string",
  games: [ gameId1, gameId2, ... ],
  friends: [ friendId1, friendId2, ... ]
}
```

Compléments d'information

- L'application doit être développée en Java.
- Le framework utilisé est libre : sparkjava, springboot, jakartaee, ...
- Ce service doit se connecter à une base de données relationnelle PostgreSQL.
- Les tests unitaires sont obligatoires.
- L'application rendue servira de support à l'entretien technique.
- La fourniture d'un fichier Dockerfile pour créer une image Docker de votre application sera appréciée.