Catan Backend API Documentation

Meriç Kerem Yalçın, Mustafa Bera Türk, Group 10

Version 1.0

1 Introduction

This document provides comprehensive documentation for the Catan Backend API, a critical component of the Catan project suite, which includes both client and server elements. The server is responsible for managing game states, user account operations, and leaderboard standings.

2 Server Subproject

The server subproject, developed as a robust Spring Boot application, consists of various layers including controllers, services, repositories, models, and configuration, each handling distinct functionalities of the application.

2.1 Controllers

Controllers serve as the entry points for HTTP requests, where they validate input, invoke service layer methods for operations, and return appropriate responses. OpenAPI annotations extensively document the API endpoints, providing a clear and interactive API specification.

2.2 Services

The service layer holds the core business logic, interfacing with repositories for data access and manipulation, and performing necessary computations and transformations to implement the application's business rules.

2.3 Repositories

Repositories simplify database interactions by offering interfaces for CRUD operations, leveraging Spring Data JPA for complex queries, which enhances development efficiency by reducing boilerplate code.

2.4 Models

Model classes, representing domain entities, are directly mapped to database tables. Entity relationships (e.g., one-to-many, many-to-many) are annotated to reflect the database schema accurately.

2.4.1 Database Relations

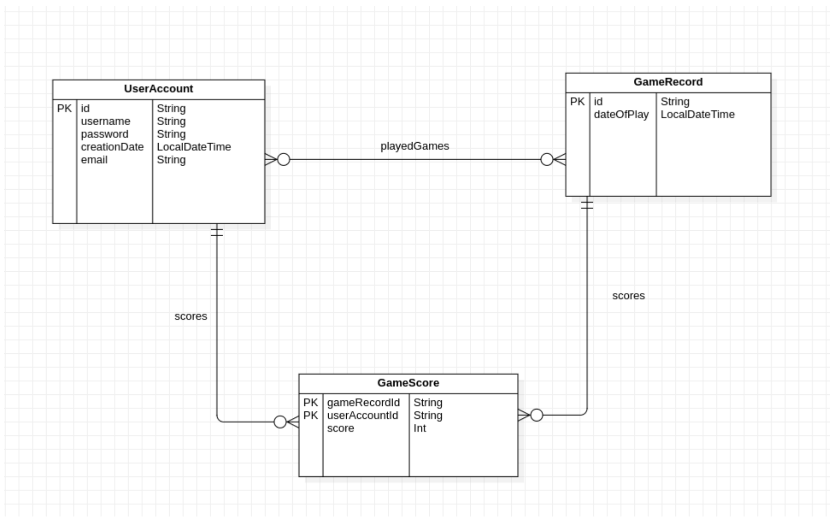
The database, designed with relational integrity, features key entities like UserAccount and GameRecord. Relationships are defined via JPA annotations, ensuring referential integrity and efficient querying.

2.5 Configuration

Configuration classes establish components essential for the application, such as OpenAPI for API documentation, SMTP for email services, and configurations for Spring Security, database connections, or custom beans.

3 Database Schema and Implementation

The database schema is crucial for understanding the relationships and data flow within the application. The ER Diagram illustrates the schema's design. The UserAccount entity, for instance, has a many-to-many relationship with the GameRecord entity, reflecting the game's nature where multiple players can participate in multiple game sessions. The GameScore entity links these two by recording scores based on user and game session.



3.1 UserAccount Entity

Represents the users of the system with attributes including a unique ID, username, password, creation date, and email.

3.2 GameRecord Entity

Represents individual game sessions with attributes including a unique ID and the date of play.

3.3 GameScore Entity

Represents the scores obtained in game sessions with attributes including a unique game record ID, user account ID, and the score as an integer.

3.4 UserAccount to GameScore Relationship

This is a one-to-many relationship, indicating that a single user can have multiple game scores, but each game score is associated with only one user.

3.5 GameRecord to GameScore Relationship

This is also a one-to-many relationship, indicating that a single game record can have multiple game scores associated with it, but each game score is related to only one game record.

3.6 UserAccount to GameRecord Relationship

This is a many-to-many relationship where users can have multiple game records and each game record can be associated with multiple users. This is because users can play multiple games and obtain scores in each, and a game can be played by multiple users each having their own scores.

4 API Endpoints and Documentation

The project employs Swagger UI for API documentation. This interactive interface allows users to understand and test various API endpoints. For instance, the GameScoreController provides endpoints for retrieving and updating game scores, while the UserAccountController handles user registration, login, and profile management. You can find our API documentation from the following link: API documentation

5 Testing Strategies

Testing is integral to our development process. The project includes a comprehensive suite of unit and integration tests. These tests cover various components of the application, including controller endpoints, service logic, and repository operations. Additionally, the H2 in-memory database is utilized for testing database interactions without affecting the production database.

6 Build and Deployment

You can find the detailed build and deployment specifications in README on our github page: Catan Backend

7 Client Subproject

The client subproject, a front-end application, which we will be working on in the near future.

8 Conclusion

This document provides a detailed overview of the Catan Backend API, covering its structure, functionalities, and development practices. It serves as a guide for developers and contributors, ensuring a clear understanding of the project's objectives and standards.