PlayerStatsApp

Project Description

This is a Spring Boot application that manages players and their statistics. It provides RESTful APIs for creating, retrieving, updating, and deleting players and their associated stats. The project uses Spring Boot, Hibernate, and Lombok for efficient development and database management.

Setup and Running Instructions

Prerequisites

- Java 17 or later
- Maven
- PostgreSQL/MySQL (or another relational database)
- An IDE (IntelliJ IDEA, Eclipse)

Installation Steps

1. **Clone the repository**

```
git clone https://github.com/your-repository.git
cd your-repository
```

2. **Configure the database**
   a. Update `application.properties` or `application.yml` with your database details:

```
spring.datasource.url=jdbc:mysql://localhost:3306/your_db name
spring.datasource.username=your_username
spring.datasource.password=your_password
spring.jpa.hibernate.ddl-auto=update
```

3. **Build and run the application**

```
mvn clean install
mvn spring-boot:run
```

The application should now be running at http://localhost:8080

API Endpoints

Player Endpoints

- **GET /players** - Retrieve a list of all players.
- **GET /players/{id}** - Get details of a specific player by ID.
- **POST /players** - Create a new player.
- **PUT /players/{id}** - Update an existing player's details.
- **DELETE /players/{id}** - Delete a player by ID.

Player Stats Endpoints

- **GET /players/{id}/stats** - Retrieve statistics for a specific player.
- **POST /players/{id}/stats** - Add statistics for a player.
- **PUT /players/{id}/stats/{Id}** - Update statistics for a player.
- **DELETE /players/{id}/stats/{Id}** - Remove statistics for a player.

Technologies Used

- Spring Boot
- Spring Data JPA
- Hibernate
- Lombok
- PostgreSQL/MySQL
- Maven