# Sports Exercise Battle (SEB)

## Overview

Sports Exercise Battle (SEB) is an HTTP/REST-based server application that gamifies push-up exercises. Users can compete against each other to see who can perform more push-ups within a 2-minute time frame. The system automatically tracks push-up counts, manages tournaments, and updates ELO ratings based on performance.

## Features

- **User Management**: Registration, login, and profile customization
- **Tournament System**: Automatic 2-minute tournaments with real-time participant tracking
- **Push-up Tracking**: Record push-up count and duration, view history and statistics
- **ELO Rating**: Competitive rating system with +2 ELO for winners, -1 for losers, +1 for ties
- **Streak System**: Track and reward consecutive days of exercise activity
- **Comprehensive Logging**: Detailed logs of tournament activities and outcomes

## Technology Stack

- **Java**: Core application language
- **TCP/HTTP**: Network communication protocols
- **Jackson**: JSON serialization/deserialization
- **PostgreSQL**: Database for persistent storage
- **JUnit**: Unit and integration testing framework

## Setup Instructions

### Prerequisites

- Java JDK 11 or higher
- Maven
- PostgreSQL

### Database Setup

1. Create a PostgreSQL database named `seb_db`
2. Create a user `webserver` with password `webserver`
3. Run the `schema.sql` script to create the necessary tables

### Build and Run

1. Clone the repository

```
git clone https://github.com/floerychristopher/sports-exercise-battle.git
cd sports-exercise-battle
```

2. Build with Maven

```
mvn clean package
```

3. Run the application

```
java -jar target/sports-exercise-battle-1.0-SNAPSHOT.jar
```

The server will start on port 10001.

# API Endpoints

## User Management

- `POST /users` - Register a new user
- `POST /sessions` - Login and get authentication token
- `GET /users/{username}` - Get user profile
- `PUT /users/{username}` - Update user profile

## Push-up Management

- `POST /history` - Record push-ups
- `GET /history` - Get push-up history

## Tournament and Stats

- `GET /tournament` - Get active tournament information
- `GET /stats` - Get user statistics
- `GET /score` - Get scoreboard

# Authentication

All endpoints except registration and login require authentication. Use the following header format:

```
Authorization: Basic username-sebToken
```

# Testing

The application includes both unit tests and integration tests:

- Run unit tests: `mvn test`
- Run integration tests: `mvn verify`

A curl test script (`seb.curl.bat`) is also provided to test all API endpoints.

# Example Usage

## Register a User

```
curl -X POST http://localhost:10001/users --header "Content-Type:
application/json" -d "{\"Username\":\"testuser\",
\"Password\":\"password123\"}"
```

### Login

```
curl -X POST http://localhost:10001/sessions --header "Content-Type:
application/json" -d "{\"Username\":\"testuser\",
\"Password\":\"password123\"}"
```

### Record Push-ups

```
curl -X POST http://localhost:10001/history --header "Content-Type:
application/json" --header "Authorization: Basic testuser-sebToken" -d
"{\"Name\": \"PushUps\", \"Count\": 40, \"DurationInSeconds\": 60}"
```

# Project Structure

```
src/
├── main/
│   ├── java/
│   │   ├── com/
│   │   │   ├── seb/
│   │   │   │   ├── Server.java
│   │   │   │   ├── RequestHandler.java
│   │   │   │   ├── config/
│   │   │   │   ├── controller/
│   │   │   │   ├── model/
│   │   │   │   ├── repository/
│   │   │   │   ├── security/
├── test/
│   ├── java/
│   │   ├── com/
│   │   │   ├── seb/
```