

## **Golf Club Tournament and Membership API**

This project is part of the **Software Design, Architecture, and Testing** course assessment, focused on Object Relational Mapping, Design Patterns, and Docker. It involves creating a functional REST API for managing golf club members and tournaments, backed by a MySQL database.

### **Features**

The API supports the following functionalities:

#### **Members**

- Add new members
- Retrieve member details
- Search members by:
  - Name
  - Membership type
  - Phone number
  - Membership start date

#### **Tournaments**

- Add new tournaments
- Retrieve tournament details
- Search tournaments by:
  - Start date
  - Location
  - Participating members

#### **Additional Functionalities**

- Add members to tournaments
- Retrieve all members participating in a specific tournament

#### **Technologies Used**

- **Backend Framework:** Spring Boot

- **Database:** MySQL
- **Containerization:** Docker
- **API Testing:** Postman

## Project Setup

### Prerequisites

- Docker installed on your machine
- Clone the project repository from GitHub

### Running the Project in Docker

1. Clone the repository:

bash

Copy code

```
git clone <repository-link>
```

```
cd <project-directory>
```

2. Build and run the Docker container:

bash

Copy code

```
docker-compose up
```

3. The API will be available at <http://localhost:8080>.

### Database Configuration

- The project uses MySQL as the database.
- The database schema and seed data will be automatically initialized when the container starts.

## Endpoints Overview

### Member Endpoints

- POST /members: Add a new member
- GET /members: Retrieve all members

- GET /members/search: Search for members based on criteria (e.g., name, membership type)

### **Tournament Endpoints**

- POST /tournaments: Add a new tournament
- GET /tournaments: Retrieve all tournaments
- GET /tournaments/search: Search for tournaments based on criteria (e.g., start date, location)

### **Member-Tournament Relationship Endpoints**

- POST /tournaments/{id}/add-member: Add a member to a tournament
- GET /tournaments/{id}/members: Retrieve members participating in a specific tournament

### **Testing**

- Use **Postman** to test API functionality. Import the provided Postman collection file for easier testing.
- Screenshots of the test cases are included in the /screenshots directory.