

后端地址: <https://git.nju.edu.cn/221250128/nju-little-fan-backend.git>

开发文档

1. 技术选型说明

1.1 数据库选型

- **数据库类型:** MySQL (基于JPA的关系型数据库)
- **原因:**
 - **数据结构复杂性:** 本项目涉及多个实体 (如 `User`, `Game`, `Team` 等), 需要处理关系复杂的数据存储, MySQL作为关系型数据库, 能很好地支持复杂的表关系和关联查询。
 - **事务管理:** MySQL支持ACID特性, 能保证数据的一致性、完整性和持久性。对于涉及到用户数据、比赛数据等敏感操作时, 事务管理尤为重要。
 - **性能和扩展性:** MySQL具有较好的性能和扩展性, 可以根据业务发展需要进行垂直或水平扩展。
 - **广泛使用:** MySQL作为一款成熟的数据库, 有大量社区支持, 能够快速解决开发过程中遇到的技术难题。

1.2 接口鉴权

- **鉴权方式:** JWT (JSON Web Token)
- **原因:**
 - **无状态性:** JWT采用无状态的认证机制, 服务端无需存储会话信息, 减少了服务器的压力, 提高了系统的扩展性。
 - **跨平台支持:** JWT能够支持多平台之间的认证与授权, 适用于不同的前后端技术栈。
 - **安全性:** JWT的签名机制保证了数据的完整性和防篡改性, 确保传输中的数据不被篡改。
 - **灵活性:** 可以根据需要在token中嵌入用户信息 (如用户ID、角色等), 便于后续的权限控制。
- **实现方式:**
 - **生成Token:** 在用户登录时, 通过密码与用户名生成JWT Token, 设置过期时间, 并将Token返回给前端。
 - **验证Token:** 在请求需要权限控制的接口时, 前端将Token传递到后端, 后端解析Token, 验证其有效性和合法性, 确保用户身份。
 - **Token过期处理:** Token过期后, 前端需要重新登录获取新的Token。

1.3 其他技术选型

- **后端框架:** Spring Boot, 用于快速开发和简化配置。
- **数据传输格式:** JSON, 作为数据交换格式, 易于理解和实现, 且具备较好的兼容性。
- **文件上传:** 阿里云OSS, 用于存储用户上传的图片或文件, 支持大文件上传和高效的分布式存储。
- **加密算法:** RSA, 采用RSA加密技术保护用户密码和敏感信息。
- **依赖管理:** Maven, 用于项目依赖管理, 确保依赖的统一版本, 简化构建过程。

2. 接口说明

2.1 游戏相关接口

- 接口路径:

```
/api/games
```

- **GET /getGameList**: 获取所有游戏信息。
- **GET /getGameById/{id}**: 根据ID获取单个游戏信息。
- **GET /getGamesByTeam/{tid}**: 根据团队ID获取该团队的所有比赛信息。
- **GET /isSubscribed/{uid}/{gid}**: 检查用户是否已订阅某个游戏。
- **POST /add**: 添加新游戏信息。

2.2 球员相关接口

- 接口路径:

```
/api/players
```

- **GET /getPlayerList**: 获取所有球员信息。
- **GET /getPlayerByTeam/{name}**: 根据团队名称获取该团队的所有球员。
- **GET /getPlayerById/{id}**: 根据ID获取单个球员信息。
- **POST /ratePlayer**: 对球员进行评分。

2.3 团队相关接口

- 接口路径:

```
/api/teams
```

- **GET /getAll**: 获取所有团队信息。
- **GET /getSuperLeagueTeam**: 获取超级联赛的所有团队。
- **GET /getTeam/{id}**: 根据ID获取单个团队信息。
- **GET /getTeamRank/{id}**: 获取团队排名。
- **GET /getChampionLeagueTeam**: 获取冠军联赛的所有团队。

2.4 用户相关接口

- 接口路径:

```
/api/users
```

- **POST /register**: 注册新用户。
- **POST /login**: 用户登录。
- **GET /**: 获取当前登录用户的信息。
- **GET /{id}**: 根据用户ID获取信息。
- **GET /setMainTeam/{uid}/{tid}**: 设置用户的主要团队。
- **GET /subscribe/{uid}/{gid}**: 用户订阅某个游戏。
- **GET /clear/{uid}/{tid}**: 清除用户的旧团队数据。
- **GET /cancel/{uid}/{gid}**: 用户取消订阅某个游戏。
- **PUT /**: 更新用户信息。
- **GET /getPublicKeyStr**: 获取RSA公钥。

