简易 QQ 聊天系统

项目简介

本项目是一个基于 Java 的简易实时聊天系统，类似于 QQ。通过 WebSocket 实现数据交互与传输，支持单用户间的通讯和群组通信。

功能特性

- **单用户聊天**：用户可以与其他单个用户进行实时聊天。
- **群组聊天**：用户可以创建群组并在群组中进行聊天。
- **消息通知**：当有新消息时，系统会进行通知。
- **用户管理**：支持用户注册、登录和管理。

技术栈

- **后端**：Java, WebSocket, MySQL
- **前端**：Java Swing (GUI)
- **依赖管理**：Maven

项目结构

```
my_qq_client/
├── src/
│ ├── main/
│ │ ├── java/cn/amatrix/
│ │ │ ├── controller/ # 控制器层，处理用户请求
│ │ │ ├── model/ # 数据模型层，定义实体类
│ │ │ ├── service/ # 服务层，包含业务逻辑
│ │ │ ├── DAO/ # 数据访问层，向数据库或后端服务器请求资源
│ │ │ ├── util/ # 工具类
│ │ │ └── Main.java # 主程序入口
│ │ └── resources/ # 图片等资源文件
│ └── test/ # 测试代码
├── target/ # 导出的 jar 文件
├── doc/ # 额外的说明文档
├── pom.xml # Maven 配置文件
└── README.md # 项目说明文件
```

安装与运行

1. **克隆项目**：

   ```
   git clone <仓库地址>
   cd <项目目录>
   ```

2. **配置数据库**：
   - 创建 MySQL 数据库并导入相关表结构。
   - 使用以下 SQL 脚本创建数据库表：

```sql
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    avatar VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    log_status ENUM('online', 'offline') DEFAULT 'offline',
    last_login_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_logout_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE friends (
    user_id INT NOT NULL,
    friend_id INT NOT NULL,
    added_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (user_id, friend_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (friend_id) REFERENCES users(user_id)
);

CREATE TABLE friend_requests (
    request_id INT AUTO_INCREMENT PRIMARY KEY,
    sender_id INT NOT NULL,
    receiver_id INT NOT NULL,
    request_message TEXT,
    request_status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',
    requested_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (sender_id) REFERENCES users(user_id),
    FOREIGN KEY (receiver_id) REFERENCES users(user_id)
);

CREATE TABLE private_messages (
    message_id INT AUTO_INCREMENT PRIMARY KEY,
    sender_id INT NOT NULL,
    receiver_id INT NOT NULL,
    message TEXT NOT NULL,
    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (sender_id) REFERENCES users(user_id),
    FOREIGN KEY (receiver_id) REFERENCES users(user_id)
);

CREATE TABLE user_groups (
```

```
    group_id INT AUTO_INCREMENT PRIMARY KEY,
    group_name VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE group_members (
    group_id INT NOT NULL,
    user_id INT NOT NULL,
    power ENUM('owner', 'admin', 'member') DEFAULT 'member',
    joined_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (group_id, user_id),
    FOREIGN KEY (group_id) REFERENCES groups(group_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE group_messages (
    message_id INT AUTO_INCREMENT PRIMARY KEY,
    group_id INT NOT NULL,
    sender_id INT NOT NULL,
    message TEXT NOT NULL,
    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (group_id) REFERENCES groups(group_id),
    FOREIGN KEY (sender_id) REFERENCES users(user_id)
);

CREATE TABLE group_join_requests (
    request_id INT AUTO_INCREMENT PRIMARY KEY,
    group_id INT NOT NULL,
    user_id INT NOT NULL,
    request_message TEXT,
    request_status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',
    requested_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (group_id) REFERENCES user_groups(group_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

- 修改 `src/main/resources/application.properties` 文件中的数据库配置。

3. **编译与运行**:

```
mvn clean install
mvn exec:java -Dexec.mainClass="cn.amatrix.Main"
```

使用说明

1. **注册与登录**：
   - 启动程序后，用户可以通过 GUI 进行注册和登录。
2. **单用户聊天**：
   - 登录后，选择联系人进行聊天。
3. **群组聊天**：
   - 创建群组并邀请成员后，可以在群组中进行聊天。

贡献

欢迎提交 Issue 和 Pull Request 来贡献代码。

许可证

本项目采用 MIT 许可证，详情请参阅 LICENSE