

“星语 (StarTalk)” 聊天室项目开发文档

1. 功能需求规格

1.1 用户模块 (User Module)

- **注册**: 用户输入手机号，系统通过第三方服务（短信宝）发送验证码，用户凭验证码完成注册。
- **登录**: 已注册用户通过手机号+验证码的方式登录。
- **用户信息**: 用户可以设置和修改自己的昵称、头像。

1.2 频道模块 (Channel Module)

- **创建频道**: 登录用户可以创建新的聊天频道。
- **创建者即管理员**: 创建频道的用户默认为该频道的管理员。
- **订阅/加入频道**: 用户可以发现并加入公开频道，或通过邀请加入私有频道。
- **消息收发**: 频道内成员可以发送和接收实时消息（文本、图片等）。

1.3 权限管理模块 (Permission Module)

- **管理员指定**: 频道创建者可以将其他成员设置或取消管理员权限。
- **禁言**: 管理员可以对频道内的某个成员进行禁言操作（在一定时间内或永久）。
- **踢出成员**: 管理员可以将成员从频道中移除。

1.4 机器人模块 (Bot Module) - 核心亮点

- **每个频道有唯一的bot**
- **Token生成**: 每个机器人生成一个唯一的认证 `token`，用于API调用。
- **智能分析**:
 - **消息总结**: 机器人可以被@调用后选择指令，对指定时间段内的频道聊天记录进行总结，并生成摘要。
 - **聊天**: 机器人可以被@调用，在不选择指令的情况下，bot将对该条聊天内容回复

- **机器人SDK**：提供一个简单易用的SDK，方便开发者（或用户）基于 `token` 快速开发自定义机器人功能。

2. 系统架构设计

代码块

```
1  graph TD
2      subgraph 用户端
3          A[前端应用 (Web/Mobile)]
4      end
5
6      subgraph 后端服务
7          B[API网关 (Gateway)]
8          C[用户服务 (User Service)]
9          D[频道服务 (Channel Service)]
10         E[消息服务 (Message Service with WebSocket)]
11         F[机器人服务 (Bot Service)]
12         G[短信发送服务 (SMS Service)]
13     end
14
15     subgraph 基础设施
16         H[消息队列 (Message Queue - 如RabbitMQ/Kafka)]
17         I[数据库 (Database - 如PostgreSQL/MongoDB)]
18         J[缓存 (Cache - 如Redis)]
19         K[短信宝API (smsbao.com)]
20     end
21
22     A --> B
23     B --> C
24     B --> D
25     B --> E
26     B --> F
27     C --> I
28     C --> J
29     D --> I
30     E --> I
31     F --> I
32     E -- 实时消息 --> A
33
34     subgraph 短信异步发送流程
35         C -- 注册/登录请求 --> H
36         G -- 消费请求 --> H
37         G -- 发送短信 --> K
38     end
```

3. 技术选型 (Tech Stack)

- 前端: Vue.js
- 后端: Springboot
- 数据库: Mysql
- 实时通信: WebSocket
- 消息队列: RabbitMQ
- 缓存: Redis (存储Session, 验证码)
- 部署: Docker

4. 数据库设计

- users表
 - id BIGINT UNSIGNED
 - user_type ENUM('human', 'bot')
 - phone_number VARCHAR
 - nickname VARCHAR
 - avatar_url VARCHAR
 - create_time DATETIME
 - update_time DATETIME
- bots表
 - id BIGINT UNSIGNED
 - bot_user_id --对应 users 表中机器人的ID
 - token VARCHAR -- 唯一的认证Token
 - create_time
 - FOREIGN KEY (bot_user_id) REFERENCES users(id)
- channels表
 - id BIGINT UNSIGNED
 - name VARCHAR

- creator_id (foreign key users.id)
 - is_public TINYINT
 - create_time
 - update_time
- channel_members表
 - id
 - channel_id
 - user_id
 - role ENUM('creator', 'admin', 'member')
 - is_muted TINYINT
 - muted_until DATETIME
 - create_time
- messages表
 - id
 - channel_id (foreign key channels.id)
 - sender_id (foreign key users.id/bots.id)
 - content_type ENUM('text', 'image', 'ai_summary')
 - content TEXT
 - metadata JSON(ai调用的元数据)
 - create_time

5. API接口设计

5.1 认证 (Auth)

路径前缀: /api/auth

功能	URL	认证	请求体	响应数据
https://msbao.com/ 发送短信验证码	POST /sendCode	否	/	验证信息
注册	POST /register	否	<div>代码块</div>	null

			<div>1 { 2 "phoneNumber": "xxx", 3 "code": "xxx" 4 }</div>	
登录	POST /login	否	<div>代码块 1 { 2 "phoneNumber": "xxx", 3 "code": "xxx" 4 }</div>	<div>代码块 1 { 2 "token": "xxx" 3 }</div>
退出登录	POST /logout	是	/	null

5.2 用户模块（User）

路径前缀： /api/users

功能	URL	认证	请求体	响应数据
获取我的信息	GET /me	是	无	<div>代码块 1 { 2 "id": "", 3 "nickname": "", 4 "avatarUrl": "" 5 }</div>
更新我的信息	PATCH /update	是	<div>代码块 1 { 2 "nickname": "xxx", 3 "avatarUrl":</div>	null

```
"xxx"
4 }
```

5.3 频道 (Channel)

路径前缀: `/api/channels`

功能	URL	认证	请求体	响应数据
创建频道	POST /	是	<div>代码块<pre>1 { 2 "name": 3 "", 4 "isPublic": 5 "" 6 }</pre></div>	null
机器人相关	POST /{channelId}/bot	是		
获取公开频道列表	GET /public	是	/	频道对象数组
获取已加入频道列表	GET /my	是	/	频道对象数组
获取频道详情	GET /{channelId}	是	/	成员列表（包括消息内容，创建时间）
加入频道	<code>POST</code> / <code>{channelId}</code> /join	是	/	null
离开频道	POST /{channelId}/leave	是	/	null
踢出成员	DELETE /{channelId}/members/{userId}	管理员	/	null
禁言成员	POST /{channelId}/members/{userId}/mute	管理员	<div>代码块<pre>1 { 2 "durationInMinutes": "" 3 }</pre></div>	null

			<div>3 }</div>	
解除禁言	DELETE /{channelId}/members/{userId}/mute	管理员	/	null
设置/取消管理员权限	PUT /{channelId}/members/{userId}/role	创建者	<div>代码块 1 { 2 "role": 3 "admin" / 4 "member" 5 }</div>	null

5.4 消息（Message）

路径前缀： /api/channels/{channelId}/messages

功能	URL	认证	请求体	响应数据
在频道发送消息	POST /	是	<div>代码块 1 { 2 "content_type": 3 "text", 4 "content": "xxx" 5 }</div>	新创建的消息对象
获取历史消息	GET /	是	请求头： 查询参数： ?page=1&limit=20	消息对象数组（分页）