javaWeb聊天室

# 项目简介

实现网页版聊天室。

功能：

后台：人员管理，cms管理

前台：登陆，注册，动态，聊天，传输文件，下载文件，踢人，管理员，授予管理员，信息修改，发邮件

# 技术分析

开发工具：eclipse

开发环境：maven

版本控制：git

运行环境：windows，Linux

后端界面：easyUI

前端技术：html/css/js/ajax

框架：ssm

运行服务器：tomcat

文件服务器：FastDFS

邮件服务器：163.com

聊天框：富文本编辑器

数据库：mysql

缓存数据库：redis

http请求处理：nginx

# 命名约定

## 项目名称

CMChatROOM

## 包命名:

Com.CM.模块.子模块….

## 变量，类命名

驼峰式

## 常量命名

大写加下划线

## 分隔符

\*@\*&

# 配置文件约定

## 常量存储

resources.properties

## 数据库信息

db.properties

## FastDFS信息

client.conf

# 服务器ip及端口

## Windows

localhost

## 开发版tomcat

CMManager:http://localhost:8080

CMRest :http://localhost:8081

CMProtal :http://localhost:8082

CMSSO :http://localhost:8083

CMSolr :http://localhost:8084

## Mysql

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/chatroom?characterEncoding=utf-8

jdbc.username=root

jdbc.password=000\*\*\*

## **Linux(CentOs)**

192.168.141.141

## Solr

http://192.168.141.141:8080

## **Redis**

http://192.168.141.141:6379

## **Fastdfs**

http://192.168.141.141: 22122

## **Nginx**

http://192.168.141.141:80

## **Git远程仓库**

http: <https://github.com/zyl-GitHub/CMChatRoom.git>

ssh: git@github.com:zyl-GitHub/CMChatRoom.git

# 数据库

|  |  |
| --- | --- |
| Chatroom | |
| User | 人员简略信息 |
| User\_desc | 人员详细信息 |
| User\_type | 人员身份信息 |
| User\_vip(放到redis中) | Vip人员 |
| Friend\_record | 好友关系记录 |
| File | 文件路径 |
| File\_category | 文件类型 |
| Content | 界面展示内容 |
| Content\_category | 界面展示内容的目录 |

## 建表

|  |  |
| --- | --- |
| User | |
| Id | 主键 自增长 |
| Account | 账号 |
| Name | 姓名 |
| Nickname | 昵称 |
| Sex | 性别 |
| Age | 年龄 |
| Phone | 电话 |
| Email | 邮箱 |
| Type\_id | 身份 关联user\_type表 |
| Sign | 个性签名 |
| Created | 创建时间 |

|  |  |
| --- | --- |
| User\_desc | |
| Id | 主键自增 |
| Hobby | 兴趣 |
| Address | 家庭住址 |
| Post | 邮编 |
|  |  |

|  |  |
| --- | --- |
| Friend\_record | |
| Id | 主键 自增长 |
| Friend\_b | 好友彼方 |
| Friend\_c | 好友此方 |

|  |  |
| --- | --- |
| User\_vip | |
| Id | 主键自增长 |
| User\_id | 用户id |

|  |  |
| --- | --- |
| File\_category | |
| Id | 主键自增 |
| FileType\_id | 文件类型代号 |
| FileType | 文件类型 |

|  |  |
| --- | --- |
| File | |
| Id | 主键自增 |
| Name | 文件名字 |
| Link | 文件路径 |
| FileType\_id | 文件类型代号 |
| Created | 文件存储时间 |

|  |  |
| --- | --- |
| Content\_category | |
| Id | 主键自增 |
| Parent\_id | 父节点id |
| Name | 名字 |
| isParent | 是否是父节点 |
| Created | 创建时间 |
| updated | 更新时间 |

|  |  |
| --- | --- |
| Content | |
| Id | 主键自增 |
| Category\_id | 分类 |
| Titile | 名字 |
| Subtitle | 子标题 |
| url | 图片路径 |
| Pic1 | 图片路径 |
| Pic2 | 图片路径 |
| Created | 创建时间 |
| updated | 更新时间 |

|  |  |
| --- | --- |
| User\_type | |
| Id | 主键自增长 |
| Type\_id | 身份代码 与id同步 |
| Description | 身份描述 |

# 项目架构

## 公共

CMCommon 工具类

CMParent jar包管理

## 后台

CM-manager

CM-manager-mapper mybatis逆向工程

CM-manager-pojo 持久化对象

CM-manager-servcie 后台服务层

CM-manager-controller 后台控制层

## 前台

表现层

CM-protal 聊天界面，功能显示

服务层

CM-solr 好友搜索

CM-sso 单点登陆系统，注册，登陆，session管理等

CM-rest 基础服务提供

# 后台管理

## 技术

框架:ssm

## 人员管理

## 根据条件查看人员列表

页面：user-list.jsp

<url:/user/quersyuser>

参数:User

返回值:EasyUIDataGridResult

### 新增人员

页面：user-add.jsp

<url:/user/save>

参数：User user

返回值:CMResult

### 编辑人员

页面：user-edit.jsp

打开

<url:/user/page/> user-edit

提交

<url:/user/update>

参数:User user

返回值:CMResult

### 删除人员

<url:/user/delete>

参数:String[] ids

返回值: CMResult

## Cms管理

**目录管理**

目录列表显示:

url: /content/category/list

参数: id Long parentId

返回值:EasyUITreeNode

目录添加:

url: /content/category/create

参数: Long parentId,String text

返回值:CMResult

删除:

url: /content/category/delete

参数: Long parentId,Integer id

返回值:CMResult

编辑

url: /content/category/update

参数: Integer id,String text

返回值:CMResult

**内容管理**

内容列表显示:

<url:/content/query/list>

参数:Integer categoryId

返回值: EasyUIDataGridResult

内容添加:

页面:

<url:/content-add>

<url:/content/save>

参数:Content content

返回值:CMResult

删除:

<url:/content/delete>

参数:String[] ids

返回值:CMResult

编辑:

页面:

[url:/content](url:/content-add)-edit

[url:/content/](url:/content/save)edit

参数:Content content

返回值:CMResult

## 3.用户登录

url:/login

页面:login.jsp

拦截器:对除/login的请求拦截，验证是否已经登陆，未登录返回/login

# 前台管理

## CM-protal

显示聊天界面

用setInterval每毫秒更新一次数据

点击用户打开新的tab

### 登陆拦截

拦截器，查询redis

### 上线

[url:/chat/user/login](file:///C:\chat\user\loginout)

参数:User user

返回值：无

### 下线

<url:/chat/user/loginout>

参数:int userId

### 检测是否还在登陆状态

<url:/chat/user/check>

参数:int userId

### 列表显示

#### 列表信息页面

<url:/chat/online/showOnline>

#### 显示在线人员列表

<url:/chat/online/activerList>

参数:无

返回值: List<EasyUITreeNode>

向rest获取userList

### 个人信息修改

页面

### 添加好友

<url:/chat/user/addFriend>

参数:int userIdb,int userIdc

返回值:CMResult

### 踢人

<url:/chat/user/delete>

参数:User user int userIdc

返回值:CMResult

### 查看用户信息

<url:/chat/user/lookInfo>

参数:int userId

返回值:CMResult

### 消息发送

<url:/chat/content/sendMessage>

参数：String message

返回值:无

### 消息显示

#### 群聊消息显示

[url:/chat/content/showAllMessage](file:///C:\chat\content\showMessage)

参数:无

返回值：String message

#### 私聊消息显示

<url:/chat/content/showPersonMessage>

参数:无

返回值：String message

### 文件上传请求

<url:/chat/upload>

参数:multipartFile uploadfile

返回值:CMResult

### 文件下载

<url:/chat/download>

参数:String link Integer id

返回值:无

### 发送邮件

### 浏览器关闭前执行下线请求

[url:/chat/user/loginout](file:///C:\chat\user\loginout)

参数:int userId

## CM-solr

数据库内容导入，搜索数据返回

### 文件搜索

<url:/solr/search/file>

参数:String query

返回值:json

### 好友搜索

<url:/solr/search/user>

参数:String query

返回值:json

## CM-sso

单点登陆系统

### 注册页面

<url:/sso/page/register>

返回注册页面

### 注册

<url:/sso/user/register>

### 销号

session管理

redis缓存

### 登陆页面

<url:/sso/page/>login

返回登陆页面

### 上线

后端从redis中获取上线列表json，jsonToList，userList添加该用户，放回到redis中

[url:/sso/user/](file:///C:\sso\page\register)login

参数:String userId String password

Key: REDIS\_SESSION\_KEY+":"+REDIS\_USER\_LOGIN\_KEY+":"+user.getUserId()

Value: user.getUserId()

返回值:CMResult

### 下线

[url:/sso/user/](file:///C:\sso\page\register)loginout

后端从redis中获取上线列表json，jsonToList，删除该用户，放回到redis中

## CM-rest

### 列表显示

<url:/rest/online/getUserList>

参数:无

返回值: List<EasyUITreeNode>

接收来自前端的请求，后端每次返回userLIst列表，前端tree定时更新

### 个人信息修改

<url:/rest/user/update>

参数:User user

返回值:CMResult

### 用户登陆检查

<url:/rest/user/check>

参数:Integer userId

返回值:CMResult

接受前端发来的前端验证该用户是否还在登录状态，从redis中获得list，比较userId

### 发送消息

#### 消息发送到后端

<url:/rest/info/sendMessage>

参数: String message

0：群聊

1：私聊

群聊

利用redis缓存，用js生成：0+useId(发送方id)+message 发送到后端

私聊

Js生成:1+userId(发送方Id)+userId(接收方Id)+message 发送到后端

#### 二、对消息进行处理

私聊：

1. hset userId(接收方id) userIdmessagecount userId(发送方Id)+message
2. 加锁
3. userIdMessagecount incr
4. Hset userId(接收方id) userIdmessagecount message

群聊:

1. 加锁
2. 从redis中获取messagecount
3. Messagecount incr
4. Hset messagecount +message

### 显示消息

<url:/rest/info/showMessage>

参数：Integer userId int mark

返回值:String message

定时向后端发送查询是否有新消息的请求

群聊 mark = 0

1. 加锁
2. 比较messagecount currentmessagecount
3. 如果不一样,currentmessagecount incr
4. 向前端发送，hget currentmessagecount userId(发送方Id)+message

私聊 mark = 1

1. 比较userIdcurrentcount和userIdmessagecount
2. 如果比较不一样，userIdcurrentcount incr,
3. 向前端发送hget userId(接收方id) userIdcurrentcount userId(发送方id)+message,
4. 收到消息处理
5. 获取userId,查看是否已经打开这个选项卡，没有打开，设置选项卡id为userId
6. 显示消息

### 撤回消息(用redis过期时间实现)

功能尚未实现

将消息放到div框中

### **文件上传**

<url:/rest/file/upload>

参数:multipartFile uploadfile

返回值:CMResult

### **文件下载**

<url:/rest/file/download>

参数:Integer id ,Integer count

### 管理员踢人

<url:/rest/user/kick>

参数:Integer userIdb Integer userIdc

返回值:CMResult

强迫下线

1. 验证权限
2. 端从redis中获取上线列表json，jsonToList，删除该用户，放回到redis中
3. 通知该用户

### 发送邮件