**详细设计规约**

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编写日期 | 说明 | 作者 | 评审时间 | 评审人员 |
| 2021/12/12 | 引言编写 | 何子硕 | 12.12 | 蒋汶霖 |
| 2021/12/13 | 系统软件结构编写 | 吴渭 | 12.14 | 蒋汶霖 |
| 2021/12/14 | 三个子系统设计 | 蒋汶霖、何子硕、李林洲 | 12.16 | 蒋汶霖、吴渭 |
| 2021/12/17 | 微服务设计 | 吴渭 | 12.17 | 何子硕 |
| 2021/12/17 | 详细类设计 | 蒋汶霖、何子硕、李林洲 | 12.19 | 李林洲、蒋汶霖 |
| 2021/12/20 | 系统部署设计 | 吴渭 | 12.20 | 蒋汶霖 |

1. **引言**

**1.1 编写目的**

详细设计说明书又可称程序设计说明书。编写的目的是说明一个软件系统结构中每个子系统或微服务或类的的设计考虑，为后序编码工作提供依据。

**1.2 背景与依据**

待开发软件系统名称：Linkedout职业发展社交平台

开发人员：蒋汶霖、何子硕、李林洲、吴渭

系统类型：Web应用

依据：需求规约、需求分析规约、概要设计规约

**1.3 参考资料**

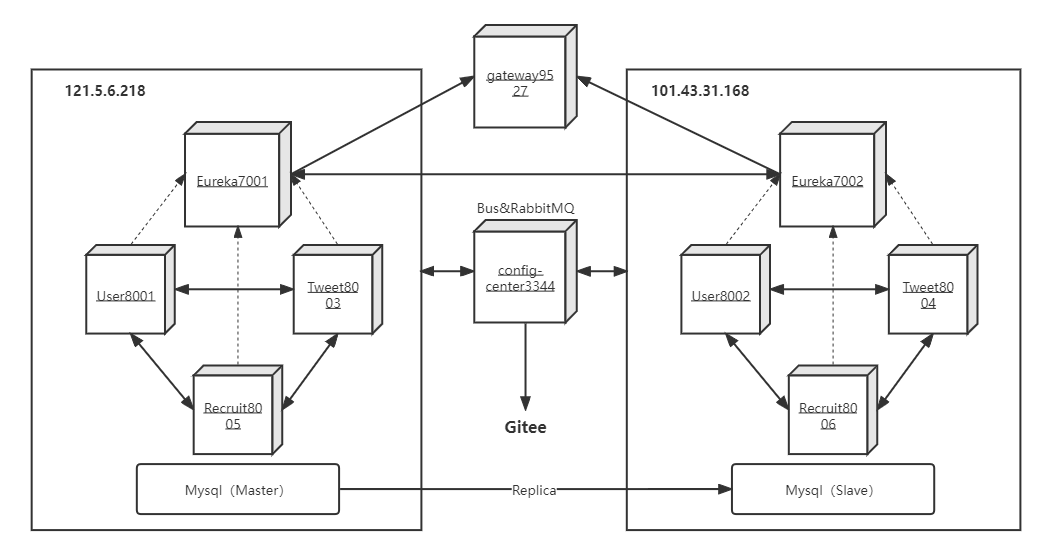
参考设计建模书籍：SoftwareEngineering-A Practitioner's Approach.8thEdition.RogerPressman.2014

GB/T 8567-2006《计算机软件文档编制规范》

1. **系统软件结构**

基于spring cloud搭建微服务架构：

* Eureka负责服务注册与治理
* 分布式部署，在101.43.31.168与121.5.68.218两台服务器上部署所有模块，通过两个Eureka相互注册，构成多节点高可用的服务集群
* OpenFeign实现服务间相互调用（自带负载均衡）
* Hystrix实现服务降级与服务熔断
* Config分布式配置中心，从Gitee仓库中实时爬取配置信息
* Bus消息总线和RabbitMQ消息队列，实现配置信息的动态刷新
* Gateway网关，对前端请求做token鉴权与日志记录，转发到服务模块的同时启用轮询负载均衡



数据架构

在两台服务器都启动了Mysql服务，采用**Mysql Replication**主从复制策略，以实现数据库层面的多节点高可用集群。MySQL主从复制涉及到三个线程，一个运行在主节点（log dump thread），其余两个(I/O thread, SQL thread)运行在从节点：

* 主节点 **binary log dump** 线程

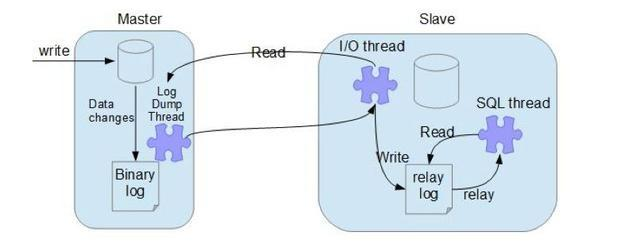
当从节点连接主节点时，主节点会创建一个log dump 线程，用于发送bin-log的内容。在读取bin-log中的操作时，此线程会对主节点上的bin-log加锁，当读取完成，甚至在发动给从节点之前，锁会被释放。

* 从节点 **I/O** 线程

当从节点上执行start slave命令之后，从节点会创建一个I/O线程用来连接主节点，请求主库中更新的bin-log。I/O线程接收到主节点binlog dump 进程发来的更新之后，保存在本地relay-log中。

* 从节点 **SQL** 线程

SQL线程负责读取relay log中的内容，解析成具体的操作并执行，最终保证主从数据的一致性。



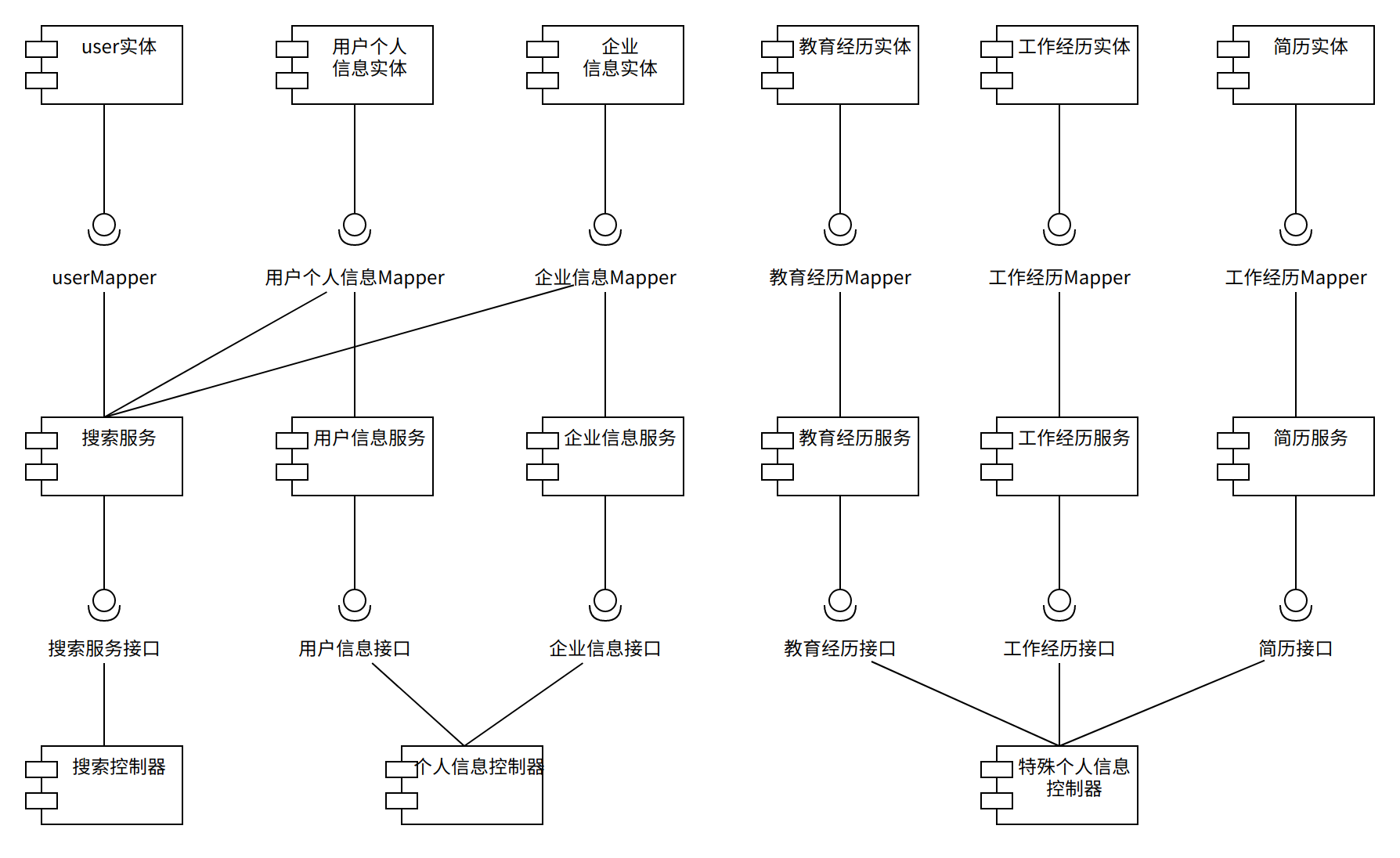
1. **子系统设计**

**3.1 用户信息子系统**

**3.1.1 子系统描述**

该子系统负责管理与平台用户相关的内容，包括基本的登录注册，个人信息的获取修改等，一些比较特殊的个人信息的获取和修改，例如教育经历，工作经历，建立文件、头像等被单独提取出来，作为控制器。用户搜索各种用户的功能也被归并到该子系统中。

**3.1.2 功能**



**3.1.3 接口**

|  |  |
| --- | --- |
| 接口名称：register |  |
| 描述 | 注册 |
| URL | POST /user/register |
| 参数 | userName：String  password：String  email：String  userType:String |
| 返回值 | 注册结果 |

|  |  |
| --- | --- |
| 接口名称：login |  |
| 描述 | 登录 |
| URL | POST /user/login |
| 参数 | userName：String  password：String |
| 返回值 | userType：String  unifiedId：String  pictureUrl：String  briefInfo：String  trueName：String |

|  |  |
| --- | --- |
| 接口名称：email |  |
| 描述 | 发送邮箱验证码 |
| URL | POST /user/email |
| 参数 | mail:String |
| 返回值 | 随机激活码 |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 维护用户个人信息 |
| URL | POST /user/userInfo |
| 参数 | unifiedId：int  trueName：String  age：int  gender：String  phoneNum：String  livePlace：String  prePosition：String  briefInfo：String |
| 返回值 | 是否修改成功 |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 获取用户个人信息 |
| URL | GET /user/userInfo |
| 参数 | uid：String  sid：String |
| 返回值 | unifiedId:int  isSubscribed：int  age：int  gender：String  phoneNum：String  prePosition：String[]  trueName:String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：userInfo |  |
| 描述 | 维护企业信息 |
| URL | POST /user/enterpriseInfo |
| 参数 | unifiedId：int  trueName：String  contactWay：String  description：String  briefInfo：String |
| 返回值 | 是否修改成功 |

|  |  |
| --- | --- |
| 接口名称：enterpriseInfo |  |
| 描述 | 获取企业信息 |
| URL | GET /user/enterpriseinfo |
| 参数 | uid：int  sid：int |
| 返回值 | unifiedId:int  isSubscribed：int  trueName:String  contactWay：String  description：String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：pic |  |
| 描述 | 上传头像 |
| URL | POST /user/pic |
| 参数 | unifiedId：int  file：File |
| 返回值 | 是否上传成功 |

|  |  |
| --- | --- |
| 接口名称：search |  |
| 描述 | 搜索企业或用户 |
| URL | GET /user/search |
| 参数 | str：String |
| 返回值 | unifiedId:int  isSubscribed：int  trueName:String  pictureUrl:String  briefInfo：String  email:String |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 增加教育经历 |
| URL | POST /user/edu |
| 参数 | unifiedId：int  startTime：String  endTime：String  collegeName：String  degree：String  major：String |
| 返回值 | 添加是否成功 |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 删除教育经历 |
| URL | DELETE /user/edu |
| 参数 | unifiedId：int  numId：int |
| 返回值 | 删除是否成功 |

|  |  |
| --- | --- |
| 接口名称：edu |  |
| 描述 | 工作经历 |
| URL | POST /user/job |
| 参数 | unifiedId：int  startTime：String  endTime：String  enterpriseName：String  positionType：String  description：String |
| 返回值 | 添加是否成功 |

|  |  |
| --- | --- |
| 接口名称：job |  |
| 描述 | 删除工作经历 |
| URL | DELETE /user/job |
| 参数 | unifiedId：int  numId：int |
| 返回值 | 删除是否成功 |

|  |  |
| --- | --- |
| 接口名称：resume |  |
| 描述 | 获取简历文件 |
| URL | GET /user/resume |
| 参数 | unifiedId：int |
| 返回值 | numId：int  resumeName：String  documentUrl：String |

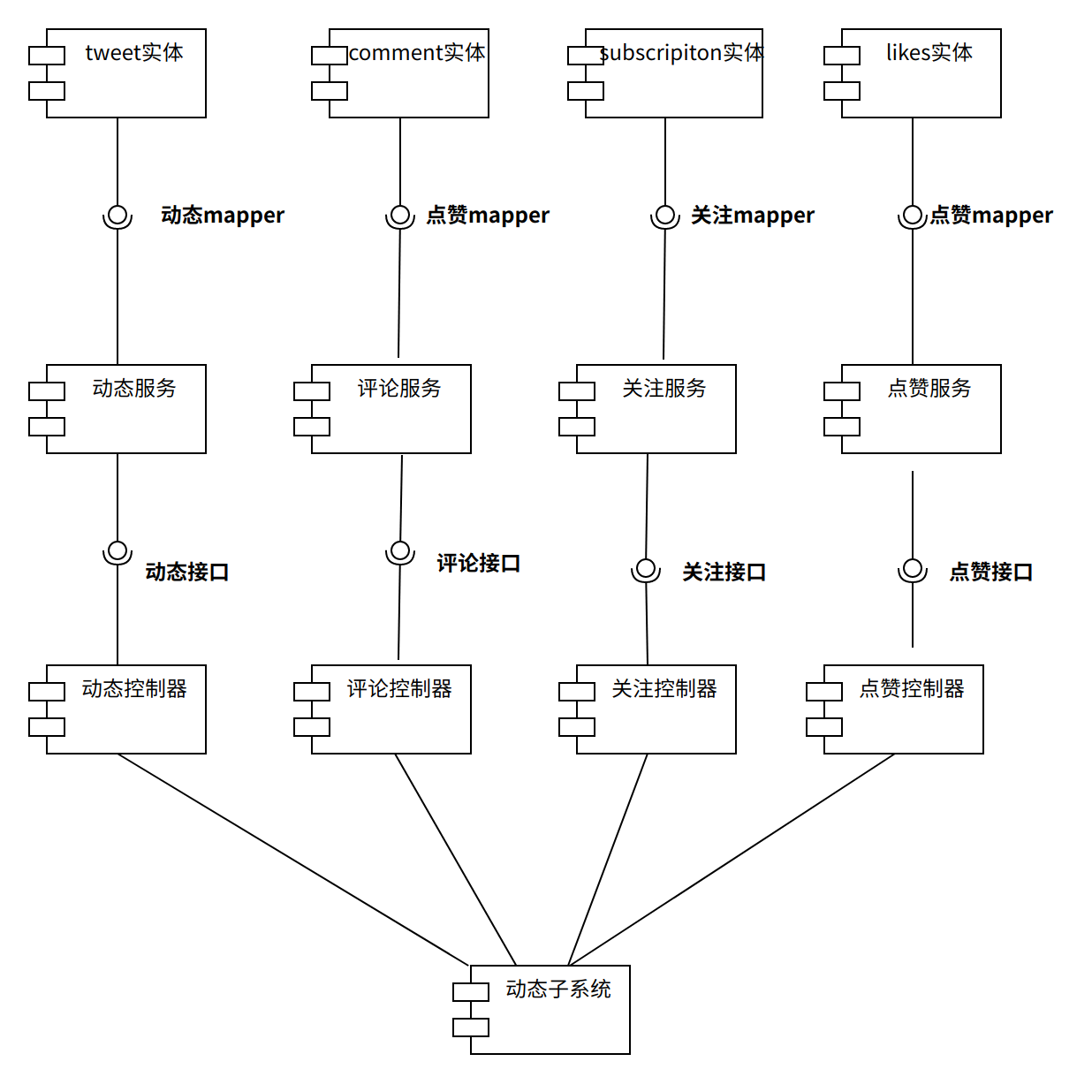
|  |  |
| --- | --- |
| 接口名称：resume |  |
| 描述 | 删除简历文件 |
| URL | DELETE /user/resume |
| 参数 | unifiedId：int  resumeId：int |
| 返回值 | 删除是否成功 |

**3.2 动态子系统**

**3.2.1 子系统描述**

动态子系统主要负责平台的动态相关服务，包括用户发布动态、评论动态、动态点赞、用户关注等，动态子系统主要部署在8003端口。

**3.2.2 功能**



**3.2.3 接口**

|  |  |
| --- | --- |
| 接口名称：putSubscription |  |
| 描述 | 关注用户 |
| URL | PUT /tweet/subscription |
| 参数 | unifiedId：int subscriptionId：int |
| 返回值 | 返回关注结果 |

|  |  |
| --- | --- |
| 接口名称：deleteSubscription |  |
| 描述 | 取消关注用户 |
| URL | DELETE /tweet/subscription |
| 参数 | unifiedId：int  subscriptionId：int |
| 返回值 | 返回取消关注结果 |

|  |  |
| --- | --- |
| 接口名称：getSubscription |  |
| 描述 | 查看是否关注用户 |
| URL | GET /tweet/subscription |
| 参数 | unifiedId：int  subscriptionId：int |
| 返回值 | 返回是否关注结果 |

|  |  |
| --- | --- |
| 接口名称：getSubscriptionList |  |
| 描述 | 获取用户关注关系列表 |
| URL | GET /tweet/subscriptionList |
| 参数 | unified：int |
| 返回值 | 返回关注关系列表 |

|  |  |
| --- | --- |
| 接口名称：getRecommend |  |
| 描述 | 获取推荐关注列表 |
| URL | GET /tweet/recommend |
| 参数 | unifiedId：int |
| 返回值 | 返回推荐用户列表 |

|  |  |
| --- | --- |
| 接口名称：putLikes |  |
| 描述 | 点赞动态 |
| URL | PUT /tweet/likes |
| 参数 | unifiedId：int  tweetId：int |
| 返回值 | 返回点赞结果 |

|  |  |
| --- | --- |
| 接口名称：deleteLikes |  |
| 描述 | 取消动态关注 |
| URL | DELETE /tweet/likes |
| 参数 | unifiedId：int  tweetId：int |
| 返回值 | 返回取消点赞结果 |

|  |  |
| --- | --- |
| 接口名称：getSelfTweet |  |
| 描述 | 查看个人动态 |
| URL | GET /tweet/self |
| 参数 | visitorId：int  intervieweeId：int  momentId：int |
| 返回值 | 返回个人动态列表 |

|  |  |
| --- | --- |
| 接口名称：getTweetList |  |
| 描述 | 获取关注者动态 |
| URL | GET /tweet/tweetList |
| 参数 | userId：int  momentId：int |
| 返回值 | 返回关注者的动态列表 |

|  |  |
| --- | --- |
| 接口名称：getComment |  |
| 描述 | 查看动态所有评论 |
| URL | GET /tweet/comment |
| 参数 | tweetId：int |
| 返回值 | 返回动态的所有评论列表 |

|  |  |
| --- | --- |
| 接口名称：postComment |  |
| 描述 | 为动态添加评论 |
| URL | POST /tweet/comment |
| 参数 | unifiedId：int  tweetId：int  content：int |
| 返回值 | 返回评论结果 |

|  |  |
| --- | --- |
| 接口名称：deleteComment |  |
| 描述 | 删除评论信息 |
| URL | DELETE /tweet/comment |
| 参数 | unifiedId：int  tweetId：int  floor：int |
| 返回值 | 返回删除评论结果 |

|  |  |
| --- | --- |
| 接口名称：PUTtweet |  |
| 描述 | 发布动态 |
| URL | PUT /tweet |
| 参数 | unifiedId：int  content：string  files：MutipartFiles |
| 返回值 | 返回新生成的动态Id |

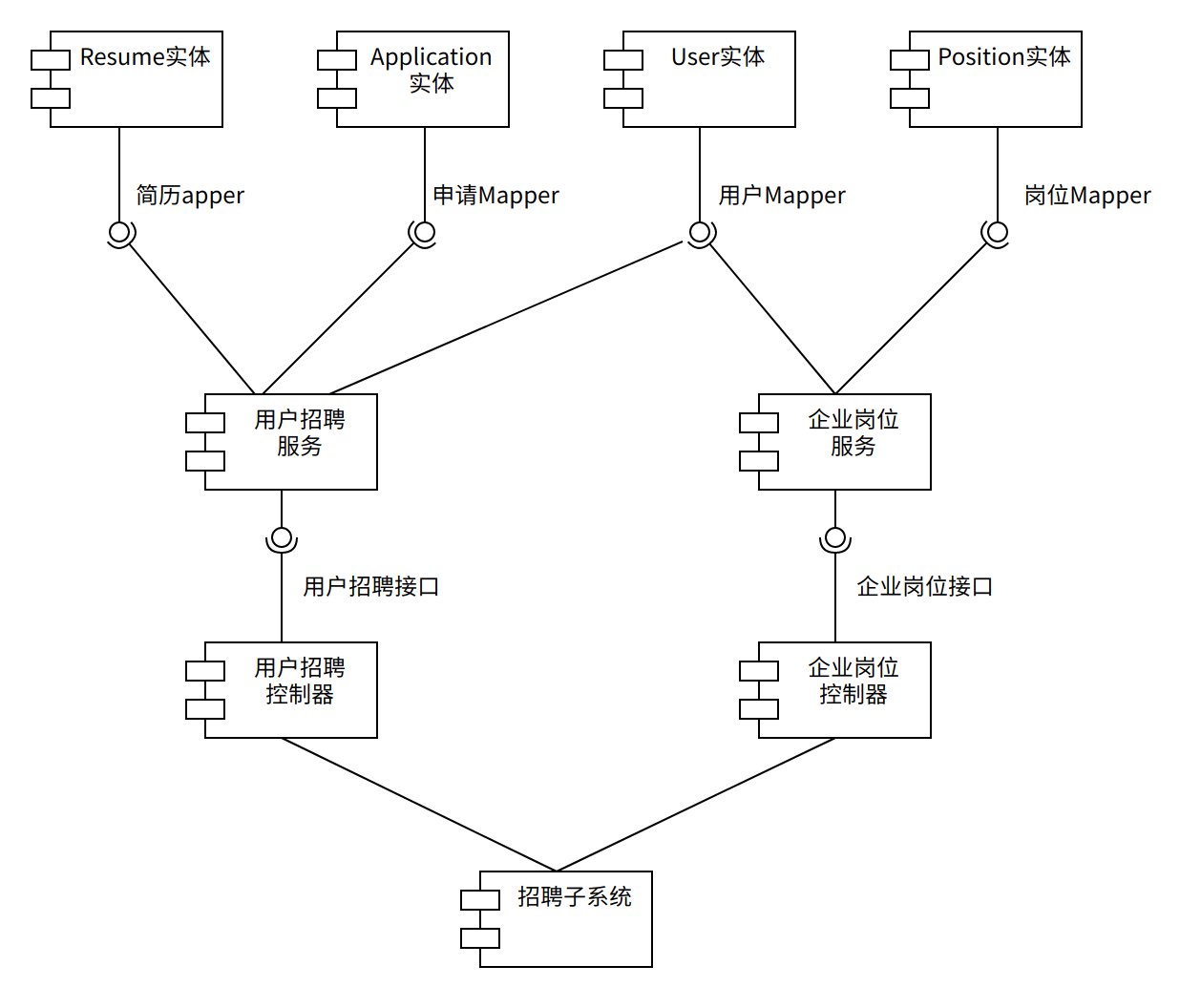
|  |  |
| --- | --- |
| 接口名称：deleteTweet |  |
| 描述 | 删除动态 |
| URL | DELETE /tweet |
| 参数 | tweetId：int |
| 返回值 | 返回删除动态结果 |

**3.3 招聘子系统**

**3.3.1 子系统描述**

招聘子系统主要包括了招聘流程相关的服务，应聘方可以查看岗位信息，获取推荐岗位，申请心仪岗位；企业可以创建岗位，编辑岗位信息，查看申请者的简历等。招聘子系统部署在8005端口

**3.3.2 功能**



**3.3.3 接口**

|  |  |
| --- | --- |
| 接口名称：postPosition |  |
| 描述 | 企业增加岗位信息 |
| URL | POST /recruit/enterprise/position |
| 参数 | unifiedId:int  description:String  reward:String  workPlace:String  contactWay:String  limitation:String  positionType:String  jobName:String |
| 返回值 | 返回增加岗位信息结果 |

|  |  |
| --- | --- |
| 接口名称：deletePosition |  |
| 描述 | 企业删除岗位信息 |
| URL | DELETE /recruit/enterprise/position |
| 参数 | unifiedId:int  jobId:int |
| 返回值 | 返回删除岗位信息结果 |

|  |  |
| --- | --- |
| 接口名称：putPosition |  |
| 描述 | 企业修改岗位信息 |
| URL | PUT /recruit/enterprise/position |
| 参数 | unifiedId:int  jobId:int  description:String  reward:String  workPlace:String  contactWay:String  limitation:String  positionType:String  jobName:String |
| 返回值 | 返回修改岗位信息结果 |

|  |  |
| --- | --- |
| 接口名称：getAllPositions |  |
| 描述 | 用户获取某家企业的所有岗位信息 |
| URL | /recruit/position/all |
| 参数 | unifiedId:int |
| 返回值 | jobId:int  reward:String  workPlace:String  positionType:String  jobName:String |

|  |  |
| --- | --- |
| 接口名称：getRecommendPositions |  |
| 描述 | 用户查看推荐岗位 |
| URL | /recruit/position/recommend |
| 参数 | unifiedId:int |
| 返回值 | unifiedId:int  briefInfo:String  pictureUrl:String  userType:int  jobId:int  reward:String  workPlace:String  positionType:String  jobName:String |

|  |  |
| --- | --- |
| 接口名称：getApplicants |  |
| 描述 | 企业查看某一岗位所有申请者信息及其简历 |
| URL | /recruit/applicants?unifiedId= &jobId= |
| 参数 | enterpriseId:int  jobId:int |
| 返回值 | unifiedId:int  briefInfo:String  pictureUrl:String  userType:int  resumeUrl:String  resumeName:String |

|  |  |
| --- | --- |
| 接口名称：getSpecifiedPosition |  |
| 描述 | 用户查看某一岗位详情 |
| URL | /recruit/position/specified |
| 参数 | unifiedId:int  jobId:int |
| 返回值 | unifiedId:int  jobId:int  description:String  reward:String  workPlace:String  contactWay:String  limitation:String  positionType:String  jobName:String  ifApplied:int |

|  |  |
| --- | --- |
| 接口名称：postApplication |  |
| 描述 | 用户发送简历/申请岗位 |
| URL | /recruit/application |
| 参数 | userId:int  enterpriseId:int  jobId:int  resumeId:int |
| 返回值 | 返回发送简历结果 |

|  |  |
| --- | --- |
| 接口名称：getApplication |  |
| 描述 | 用户查看投递过的所有岗位 |
| URL | /recruit/application |
| 参数 | userId |
| 返回值 | unifiedId:int  briefInfo:String  pictureUrl:String  userType:int  jobId:int  reward:String  workPlace:String  positionType:String  jobName:String |

1. **微服务设计**

**4.1 微服务功能**

该项目采用分布式微服务的架构以应对未来可能的高并发量，为了实现微服务的分模块部署，需要将系统合理的拆成多个模块，从而将它们分布式地部署在不同的服务器上。但由于资金条件限制，我们未能将不同模块部署在多台服务器上，当前部署方案如本文档第2点中第一幅图所示。其中用户个人信息子系统、动态子系统、招聘子系统均复制了两份部署在两台服务器上，在系统发生故障时仍然能够正常进行，防止一个模块的崩溃波及整个系统。所有模块均前往配置中心拉取，数据库服务器访问地址的信息。数据库也存有两份，一个主数据库，一个从数据库，通过采用Mysql Replication的主从复制策略，以实现数据库层面的多节点高可用集群。

**4.2 和哪些微服务有交互**

用户个人信息模块是是整个系统的核心模块，它将为动态和招聘模块提供相应需要的用户信息。因此我们设计时，对于需要这两个模块中需要用到个人信息子系统的信息时，设计为对用户个人信息子系统的接口调用，而不是直接对数据库进行操作，下列表格展现了哪些动态和招聘子系统的接口将会调用用户信息子系统的API。

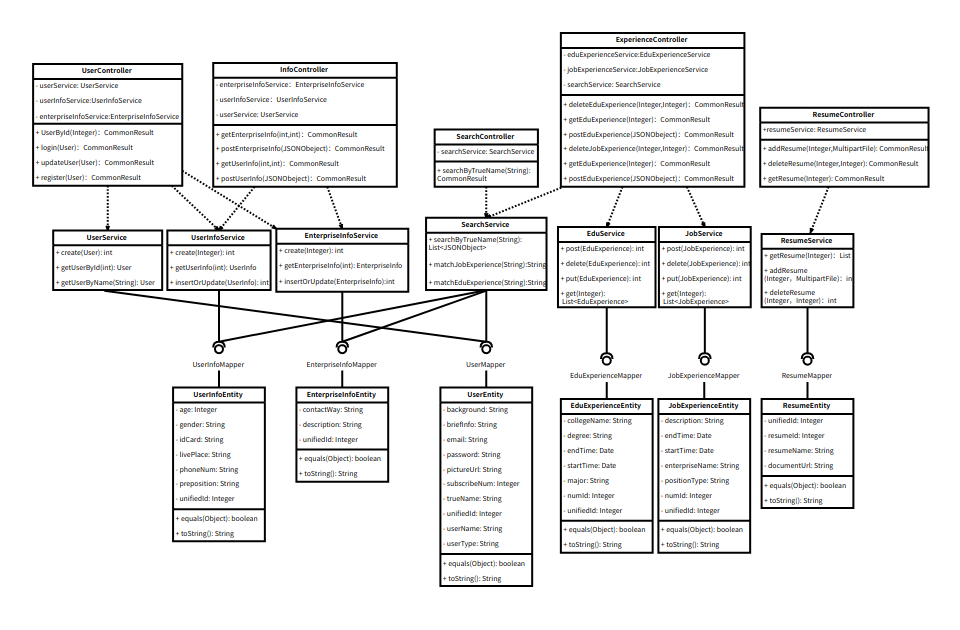
|  |  |  |
| --- | --- | --- |
| 模块名称 | 相关接口 | 实现描述 |
| Tweet | /tweet/recommend/{unifiedId} | 推荐关注的返回的列表中，应该包含推荐用户的基本信息，因此调用获取个人信息接口 |
|  | /tweet/tweetList | 查看关注者动态，需要知道该条动态所属用户的头像、姓名等信息，因此需要调用获取个人信息接口 |
|  | /tweet/comment/{tweetId} | 评论和动态同理需要知道评论者的个人信息 |
| recruitment | /recruit/applicants | 企业查看某一岗位所有申请者信息及其简历时，用户的个人信息以及简历均来自与个人信息子系统，因此，该模块无需直接查询底层数据库，直接调用个人信息子系统的接口即可 |
|  | recruit/position/recommend | 用户查看推荐岗位与用户查看推荐关注类似，除了需要返回推荐岗位信息，还应返回该岗位所属企业账号的相关信息 |
|  | recruit/application | 用户查看投递过的所有岗位时，除了需要知道投递到了哪个具体的岗位，还需要知道该岗位所属企业的相关信息 |

1. **类设计**

**5.1 用户信息子系统**

**5.1.1 功能**

用户子系统中底层需要和存储在数据库中的各种信息实体类打交道，通过Mapper接口，服务层来间接与数据库通信，一些服务仅需要与一个实体类通信，搜索服务需要与多个数据库实体交互。最顶层的controller层调用服务层提供的服务，类似功能的服务交由一个controller进行转发即可



**5.1.2 输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：EduExperienceService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postEduExperience | EduExperience | int | 添加教育经历 |
| deleteEduExperience | Integer unifiedId  Integer numId | int | 删除教育经历 |
| putEduExperience | EduExperience | int | 修改教育经历 |
| getEduExperience | Integer unifiedId | List<EduExperience> | 获取教育经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：EnterpriseInfoService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | Integer unifiedId | int | 创建企业信息 |
| getEnterpriseInfo | Int unifiedId | EnterpriseInfo | 获取企业信息 |
| insertOrUpdateEnterpriseInfo | EnterpriseInfo | int | 插入或更新企业用户信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：JobExperienceService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postJobExperience | JobExperience | int | 添加工作经历 |
| deleteJobExperience | Integer unifiedId  Integer numId | int | 删除工作经历 |
| putJobExperience | JobExperience | int | 修改工作经历 |
| getJobExperience | Integer unifiedId | List<JobExperience> | 获取工作经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：ResumeService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| getResume | Integer unifiedId | List<Resume> | 获取简历 |
| addResume | Integer unifiedId  MultipartFile | int | 添加新简历 |
| deleteResume | Integer unifiedId  Integer ResumeId | int | 删除简历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：HeadPictureService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| addHeadPicture | Integer unifiedId  MultipartFile file | int | 上传头像 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：SearchService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| searchByTrueName | String str | List<JSONObeject> | 按照用户真实姓名搜索 |
| matchJobExperience | String str | String | 匹配工作经历 |
| matchEduExperience | String str | String | 匹配教育经历 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserInfoService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | Integer unfiedId | int | 创建用户个人信息 |
| getUserinfo | Int unifiedId | Userinfo | 获取用户个人信息 |
| inserOrUpdateUserInfo | UserInfo | int | 更新或插入用户个人信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| create | User | int | 创建用户 |
| getUserById | Int unifiedId | User | 根据统一Id获取用户 |
| getUserByName | String username | User | 根据用户名获取用户 |

**5.1.3 属性详细描述**

|  |  |
| --- | --- |
| class：EduexperienceServiceImplement |  |
| 简介：教育经历服务实现类 |  |
| 属性 | 描述 |
| eduExperienceMapper | 教育经历数据访问对象 |

|  |  |
| --- | --- |
| class：EnterpriseInfoServiceImplement |  |
| 简介：企业信息服务实现类 |  |
| 属性 | 描述 |
| enterpriseInfoMapper | 企业信息数据访问对象 |

|  |  |
| --- | --- |
| class：HeadPictureServiceImplement |  |
| 简介：头像服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

|  |  |
| --- | --- |
| class：JobExperienceServiceImplement |  |
| 简介：工作经历服务实现类 |  |
| 属性 | 描述 |
| jobExperienceMapper | 工作经历数据访问对象 |

|  |  |
| --- | --- |
| class：MailServiceImplement |  |
| 简介：邮件服务实现类 |  |
| 属性 | 描述 |
| JavaMailSender | java邮件发送对象 |

|  |  |
| --- | --- |
| class：ResumeServiceImplement |  |
| 简介：简历服务实现类 |  |
| 属性 | 描述 |
| resumeMapper | 简历数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

|  |  |
| --- | --- |
| class：SearchServiceImplement |  |
| 简介：搜索服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |

|  |  |
| --- | --- |
| class：UserInfoServiceImplement |  |
| 简介：用户个人信息服务实现类 |  |
| 属性 | 描述 |
| userInfoMapper | 用户个人信息数据访问对象 |

|  |  |
| --- | --- |
| class：UserServiceImplement |  |
| 简介：用户服务实现类 |  |
| 属性 | 描述 |
| userMapper | 用户数据访问对象 |
| bucketName | oss存储服务名 |
| endpoint | oss存储服务端口 |

**5.1.4 方法详细描述及实现流程**

|  |  |  |
| --- | --- | --- |
| class：EduexperienceServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| postEduExperience | 添加教育经历 | 根据前端返回的教育经历信息，向数据库中插入一条新的教育经历 |
| deleteEduExperience | 删除教育经历 | 根据前端返回的统一id，删除数据库中的指定教育经历 |
| putEduExperience | 修改教育经历 | 根据前端返回的统一id和教育经历信息，更新数据库已有教育经历 |
| getEduExperience | 获取教育经历 | 根据前端返回的统一id获取该用户的所有教育经历 |

|  |  |  |
| --- | --- | --- |
| class：EnterpriseInfoServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建企业信息 | 向数据库的EnterpriseInfo表中插入新加入的企业信息，返回统一id |
| getEnterpriseInfo | 获取企业信息 | 根据前端传入的统一id，去数据库中查找该id对应的企业信息，并返回该企业信息 |
| insertOrUpdateEnterpriseInfo | 插入或更新企业用户信息 | 根据前端传入的id，去数据库中查找该id对应的企业信息是否存在，找到后用前端传入的信息更新该企业信息 |

|  |  |  |
| --- | --- | --- |
| class：HeadPictureServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| addHeadPicture | 上传头像 | 将用户上传的建立文件通过java类MuiltPartFile存储到指定位置，并向数据库中插入对应位置的url |

|  |  |  |
| --- | --- | --- |
| class：JobExperienceServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| postJobExperience | 添加工作经历 | 根据前端返回的工作经历信息，向数据库中插入一条新的工作经历 |
| deleteJobExperience | 删除工作经历 | 根据前端返回的统一id，删除数据库中的指定工作经历 |
| putJobExperience | 修改工作经历 | 根据前端返回的统一id和工作经历信息，更新数据库已有工作经历 |
| getJobExperience | 获取工作经历 | 根据前端返回的统一id获取该用户的所有工作经历 |

|  |  |  |
| --- | --- | --- |
| class：MailServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| sendMail | 发送随机激活码邮件 | 调用邮件发送相关服务，向前端传入的邮箱地址发送一封带有随机验证码的邮件。 |

|  |  |  |
| --- | --- | --- |
| class：ResumeServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| getResume | 获取简历 | 根据前端传入的统一id，返回该用户上传的所有简历的存储的url |
| addResume | 添加新简历 | 将用户上传的建立文件通过java类MuiltPartFile存储到指定位置，并向数据库中插入对应位置的url |
| deleteResume | 删除简历 | 根据前端传入的统一id和简历id删除对应用户的指定建立文件，同时删除数据库中的记录 |

|  |  |  |
| --- | --- | --- |
| class：SearchServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| searchByTrueName | 按照用户真实姓名搜索 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的真实姓名 |
| matchJobExperience | 匹配工作经历 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的工作公司 |
| matchEduExperience | 匹配教育经历 | 根据传入的搜索字段，利用余弦匹配算法，返回用户可能搜索的高校名称 |

|  |  |  |
| --- | --- | --- |
| class：UserInfoServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建用户个人信息 | 向数据库的UserInfo表中插入新加入的用户信息，返回统一id |
| getUserinfo | 获取用户个人信息 | 根据前端传入的统一id，去数据库中查找该id对应的用户个人信息，并返回该用户个人信息 |
| inserOrUpdateUserInfo | 更新或插入用户个人信息 | 根据前端传入的id，去数据库中查找该id对应的用户个人信息是否存在，找到后用前端传入的信息更新该用户个人信息 |

|  |  |  |
| --- | --- | --- |
| class：UserServiceImplement |  |  |
| 方法名 | 描述 | 实现 |
| create | 创建用户 | 向数据库的User表中插入新加入的用户信息，设定用户类型，返回统一id |
| getUserById | 根据统一Id获取用户 | 根据前端传入的统一id，去数据库中查找该id对应的用户，并返回该用户信息 |
| getUserByName | 根据用户名获取用户 | 根据前端传入的用户名，去数据库中查找该用户名对应的用户，并返回该用户信息 |

**5.1.5 限制条件及出错处理**

|  |  |
| --- | --- |
| class：EduExperienceController |  |
| 方法名 | 描述 |
| postEduExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| deleteEduExperience | 判断传入的统一认证id和顺序id是否存在，否则返回失败 |
| putEduExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| getEduExperience | 判断传入的统一认证i是否存在，否则返回失败 |

|  |  |
| --- | --- |
| class：EnterpriseInfoController |  |
| 方法名 | 描述 |
| getEnterpriseInfo | 判断查看者与被查看者的id是否相同 |
| updateEnterpriseInfo | 更新前需判断用户是否存在，并且传入的id不能为空 |

|  |  |
| --- | --- |
| class：HeadPictureController |  |
| 方法名 | 描述 |
| addHeadPicture | 检测是否存在上传头像的用户id，上传文件需捕获异常，防止文件上传失败 |

|  |  |
| --- | --- |
| class：JobExperienceController |  |
| 方法名 | 描述 |
| postJobExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| deleteJobExperience | 判断传入的统一认证id和顺序id是否存在，否则返回失败 |
| putJobExperience | 转换时间格式字符串时，使用异常捕捉，避免不符合格式的时间无法转化导致报错。判断传入参数是否包含统一认证id，否则报错 |
| getJobExperience | 判断传入的统一认证i是否存在，否则返回失败 |

|  |  |
| --- | --- |
| class：ResumeController |  |
| 方法名 | 描述 |
| getResume | 判断传入参数是否包含统一认证id，否则报错 |
| addResume | 判断传入参数是否包含统一认证id。上传文件需捕获异常，防止文件上传失败 |
| deleteResume | 判断是否存在对应的统一id和简历id对应的数据库数据，没有则报删除失败 |

|  |  |
| --- | --- |
| class：SearchController |  |
| 方法名 | 描述 |
| searchTrueName | 判断关键字调用搜索服务后，搜索结果是否为空 |

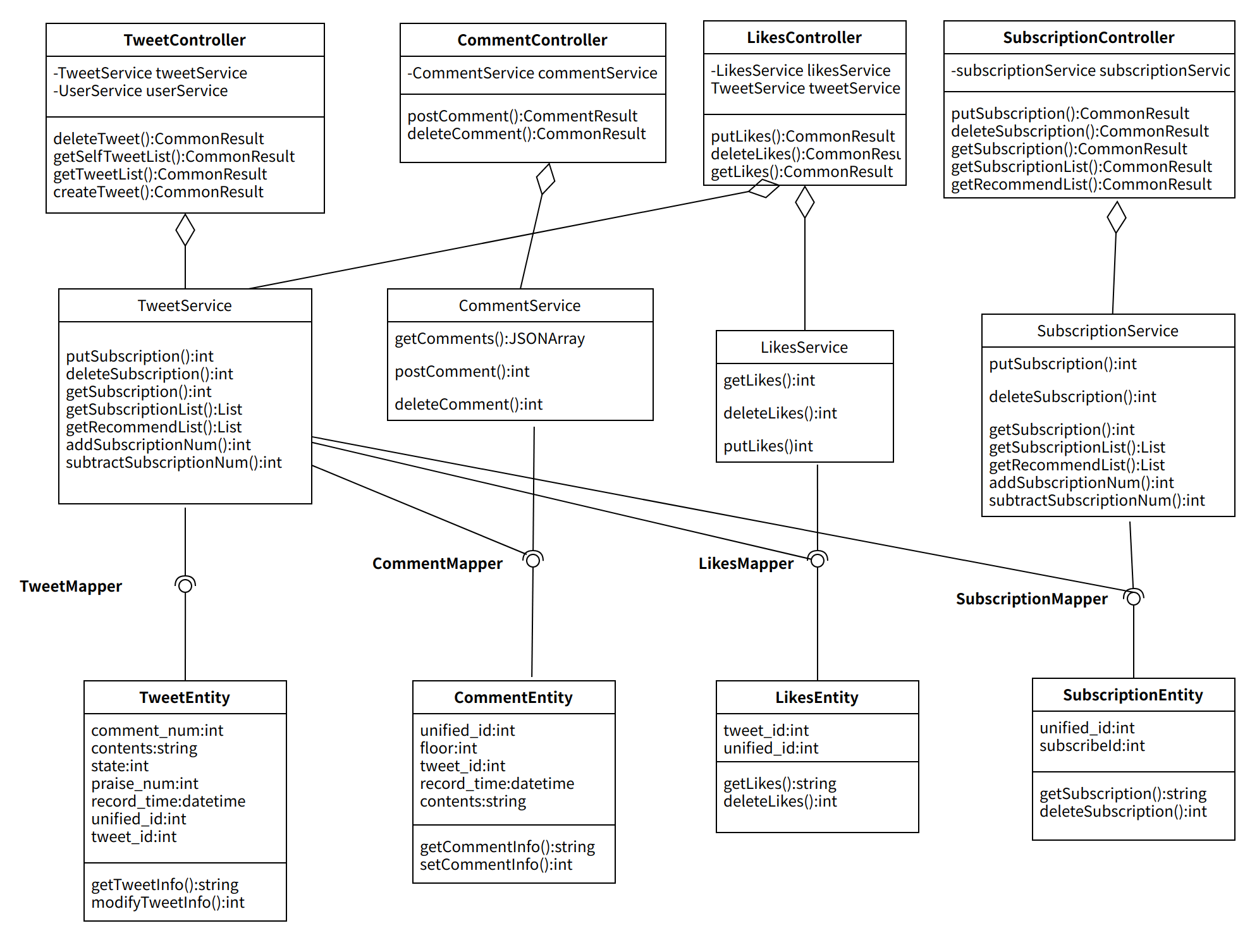
|  |  |
| --- | --- |
| class：UserController |  |
| 方法名 | 描述 |
| create | 判断用户名是否有重复，判断前端传入的用户类型是否合法 |
| getMailCaptcha | 判断同一邮箱的请求次数，多次调用触发服务熔断 |
| login | 判断用户的密码是否正确后才执行cookie、token生成操作 |
| UserById | 判断查询的返回结果是否有内容 |
| updateUser | 判断传入参数是否包含统一认证id，否则报错 |

|  |  |
| --- | --- |
| class：UserInfoController |  |
| 方法名 | 描述 |
| getUserInfo | 判断查看者与被查看者的id是否相同 |
| updateUserInfo | 更新前需判断用户是否存在，并且传入的id不能为空 |

**5.2 动态子系统**

**5.2.1 功能**

动态子系统通过各个位于顶层的Controller来处理外界请求，Controller层主要面向前端的访问请求。Controller层主要通过调用Service的各个方法来处理请求，Service层的功能是将Controller细化过的功能进行实现，通过调用不同的Mapper接口来间接操作数据库，从而实现数据的增删改查操作。



**5.2.2 输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：CommentService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| postComment | Int tweetId | int | 输入参数为动态的id  返回值为请求结果 |
| deleteComment | Int unifiedId  Int tweetId  Int floor | int | 输入参数为动态id、用户id和楼层数  返回值为删除评论结果 |
| getComments | Int tweetId | JSONArray | 输入值为动态id  返回值为评论列表 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：LikesService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| getLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为点赞查询结果 |
| deleteLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为取消点赞结果 |
| putLikes | Int unifiedId  Int tweetId | int | 输入参数为用户id和动态id  返回值为点赞结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：SubscriptionService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| putSubscription | Int unifiedId  Int subscriptionId | Int | 输入参数为用户id和被关注用户id  返回值为关注结果 |
| deleteSubscription | Int unifiedId  Int subscriptionId | Int | 输入参数为用户id和被关注用户id  返回值为取消关注结果 |
| getSubscription | Int unifiedId  Int subscriptionId | int | 输入参数为用户id和被关注用户id  返回值为关注结果 |
| getSubscriptionList | Int unifiedId | List | 输入参数为用户id  返回值为关注列表 |
| getRecommendList | Int unifiedId | List | 输入参数为用户id  返回值为推荐用户列表 |
| addSubscriptionNum | Int unifiedId | int | 输入参数为用户id  返回值为请求结果 |
| subtractSubscriptionNum | Int unifiedId | int | 输入参数为用户id  返回值为请求结果 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：TweetService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| addLikesNum | Int tweetId | int | 输入参数为动态id  返回值为点赞结果 |
| subtractLikesNum | Int tweetId | int | 输入参数为动态id  返回值为取消点赞结果 |
| deleteTweet | Int tweetId | int | 输入参数为动态id  返回值为删除动态结果 |
| getSeltTweetList | Int visitorId  Int intervieweeId  Int momentId | JSONArray | 输入参数为访问者id、被访问者id、当前浏览最后动态id  返回值为个人动态列表 |
| getTweetList | Int visitorId  Int momentId | JSONArray | 输入参数为访问者id和当前浏览最后动态id  返回值为关注者的动态列表 |
| getSimpleUserInfo | Int unifiedId | JSONArray | 输入参数为用户id  返回值为用户的简要信息 |
| getTweetPicture | Int tweetId | List | 输入参数为动态id  返回值为动态的图片列表 |
| createTweet | Int unifiedId  String content  Date recordTime  List files | int | 输入参数为用户id、动态内容、发布时间、动态图片列表  返回值为动态的id |
| uploadPic | Int unifiedId  List files | int | 输入参数为动态id、动态图片  返回值为存储结果 |
| maxTweetId | none | int | 无输入参数  返回值为最大的动态id |

**5.2.3 属性详细描述**

|  |  |
| --- | --- |
| class：CommentServiceImpl |  |
| 简介：评论服务接口实现类 |  |
| 属性 | 描述 |
| CommentMapper  commentMapper | 评论DAO层 |
| TweetMapper  tweetMapper | 动态DAO层 |
| TweetService  tweetService | 动态相关服务类 |

|  |  |
| --- | --- |
| class：LikesServiceImpl |  |
| 简介：点赞服务接口实现类 |  |
| 属性 | 描述 |
| LikesMapper  likesMapper | 点赞DAO层 |
| TweetMapper  tweetMapper | 动态DAO层 |

|  |  |
| --- | --- |
| class：SubscriptionServiceImpl |  |
| 简介：关注服务接口实现类 |  |
| 属性 | 描述 |
| SubscriptionMapper  subscriptionMapper | 关注DAO层 |
| UserMapper  userMapper | 用户DAO层 |

|  |  |
| --- | --- |
| class：TweetServiceImpl |  |
| 简介：动态服务接口实现类 |  |
| 属性 | 描述 |
| TweetMapper  tweetMapper | 动态DAO层 |
| LikesMapper  likesMapper | 点赞DAO层 |
| UserMapper  userMapper | 用户DAO层 |
| PictureMapper  pictureMapper | 动态图片DAO层 |
| SubscriptionMapper  subscriptionMapper | 关注DAO层 |

**5.2.4 方法详细描述及实现流程**

|  |  |  |
| --- | --- | --- |
| class：CommentServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| postComment | 发布动态评论 | 根据前端返回的评论信息，在数据库中插入一条新的评论信息 |
| deleteComment | 删除动态评论 | 根据相应动态id、用户id和楼层数，在数据库中找到相应记录进行删除 |
| getComments | 获取动态评论 | 根据相应动态id，在数据库中查询相应id的动态的所有评论 |

|  |  |  |
| --- | --- | --- |
| class：LikesServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| getLikes | 查询点赞状态 | 根据相应用户id和动态id，在数据库中查询是否存在点赞关系 |
| deleteLikes | 取消动态点赞 | 根据相应用户id和动态id，从数据库中找到相应记录进行删除 |
| putLikes | 动态点赞 | 通过相应用户id和动态id，在数据库中新建一条点赞记录 |

|  |  |  |
| --- | --- | --- |
| class：SubscriptionServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| putSubscription | 关注用户 | 根据相应用户id和被关注者id，在数据库中插入一条新的关注记录 |
| deleteSubscription | 取消关注用户 | 根据用户id和被关注者id，在数据库中找到相应记录进行删除 |
| getSubscription | 查询是否关注某用户 | 根据用户id和被关注者id，在数据库中查找是否存在相应记录 |
| getSubscriptionList | 查询用户的所有关注者信息 | 根据相应用户id，在数据库中查找该用户的所有关注列表信息 |
| getRecommendList | 向用户推荐其他未关注的用户 | 根据相应用户id，查找该用户尚未关注的用户信息进行返回 |
| addSubscriptionNum | 增加用户被关注数 | 根据相应用户id，在数据库中寻找该用户信息，将其被关注数加一 |
| subtractSubscriptionNum | 减少用户被关注数 | 根据相应用户id，在数据库中寻找该用户信息，将其被关注数减一 |

|  |  |  |
| --- | --- | --- |
| class：TweetServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| addLikesNum | 增加动态点赞数 | 根据相应动态id，在数据库中找到相应动态信息，将其点赞数加一 |
| subtractLikesNum | 减少动态点赞数 | 根据相应动态id，在数据库中找到相应动态信息，将其点赞数减一 |
| deleteTweet | 删除动态 | 根据相应动态id，在数据库中找到该动态信息进行删除操作 |
| getSeltTweetList | 获取单独用户的所有动态 | 根据相应用户id，在数据库中寻找该用户发布的所用动态，根据momentId，返回比该编号小的十条动态 |
| getTweetList | 获取用户关注者的动态 | 根据相应用户id，在数据库中寻找该用户所有关注者发布的所用动态，根据momentId，返回比该编号小的十条动态 |
| getSimpleUserInfo | 获取用户简单个人信息 | 根据相应用户id，在数据库中查询该用户的简单用户信息 |
| getTweetPicture | 获取动态的所有图片 | 根据相应动态id，在数据库中查询该动态的所有动态图片 |
| createTweet | 发布动态 | 根据相应的动态内容，在数据库中新生成一条动态信息，并将生成的动态信息编号进行返回 |
| uploadPic | 上传动态图片至服务器 | 根据相应动态id，在数据库中新生成一条图片存储记录，并将图片存至服务器指定位置 |
| maxTweetId | 查询数据库中动态的最大编号 | 寻找当前数据库中动态信息的最大编号 |

**5.2.5 限制条件及出错处理**

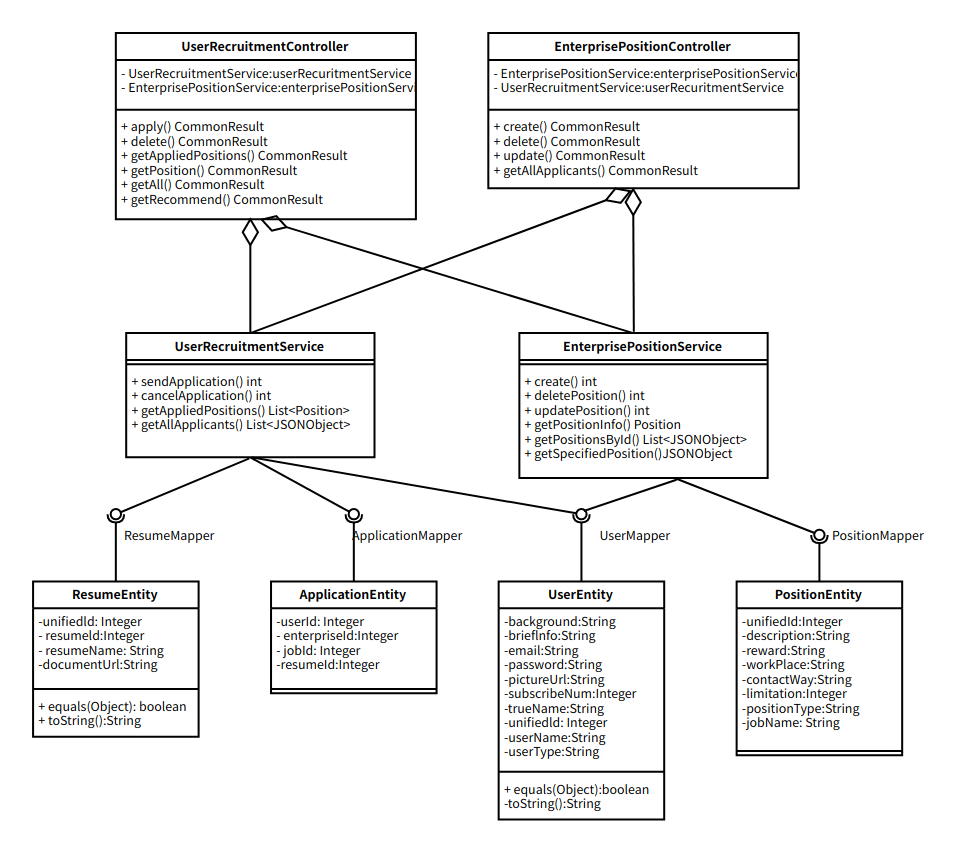
|  |  |
| --- | --- |
| class：TweetServiceImpl |  |
| 方法名 | 描述 |
| uploadPic | 在调用该方法向服务器上传图片时，若上传发生错误，则会抛出一个IOException |

|  |  |
| --- | --- |
| class：CommentServiceImpl |  |
| 方法名 | 描述 |
| postComment | 在调用该方法时，新的评论的楼层数会是原来的最大楼层数加一。若原先最大楼层数为null(没有评论)，此时该评论的楼层数将被设置为1 |

**5.3 招聘子系统**

**5.3.1 功能**

招聘子系统的视线主要包括三个层次，最上层是Controller层，这里包含两个Controller，分别负责用户招聘以及企业岗位相关的功能，是接口与实现之间的桥梁。Controller下方是Service，主要负责功能的实现，再下方是与数据库交互的DAO层，这里我们定义为若干个实体，服务如果要与数据库交互，那么需要通过不同的Mapper来实现。



**5.3.2 输入输出项**

|  |  |  |  |
| --- | --- | --- | --- |
| class：EnterprisePositionService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| isEnterprise | Int unifiedId | Boolean | 根据id判断某一个用户是否是企业用户 |
| isCommonUser | Int unifiedId | Boolean | 根据id判断某一个用户是否是普通用户 |
| create | Position position | int | 创建岗位 |
| deletePosition | int unifiedId, int jobId | int | 删除岗位 |
| updatePosition | Position position | int | 更新岗位 |
| getPositionInfo | int unifiedId, int jobId | Position | 获取该企业创建的岗位信息 |
| getSpecifiedPosition | int unifiedId, int jobId | JSONObject | 获取指定岗位信息+用户是否申请该岗位 |
| getPositionsById | int unifiedId,Integer momentId | List<JSONObject> | 获取企业的所有岗位 |
| getRecommendedPositionsById | int unifiedId,Integer momentId | List<JSONObject> | 根据id获取用户的推荐岗位 |

|  |  |  |  |
| --- | --- | --- | --- |
| class：UserRecruitmentService |  |  |  |
| 方法名 | 参数 | 返回值 | 描述 |
| sendApplication | Application application | int | 用户发送岗位申请 |
| cancelApplication | Application application | int | 用户取消岗位申请 |
| getAppliedPositions | int userId | List<Position> | 用户获取申请过的所有岗位 |
| getAllApplicants | int unifiedId, int jobId | List<JSONObject> | 企业查看岗位所有申请者信息 |

**5.3.3 属性详细描述**

|  |  |
| --- | --- |
| class：EnterprisePositionServiceImpl |  |
| 简介：企业岗位服务接口实现类 |  |
| 属性 | 描述 |
| PositionMapper | 岗位DAO层 |
| UserMapper | 用户DAO层 |

|  |  |
| --- | --- |
| class：UserRecruitmentServiceImpl |  |
| 简介：招聘服务接口实现类 |  |
| 属性 | 描述 |
| ApplicationMapper | 岗位申请DAO层 |
| UserMapper | 用户DAO层 |
| ResumeMapper | 简历DAO层 |

**5.3.4 方法详细描述及实现流程**

|  |  |  |
| --- | --- | --- |
| class：EnterprisePositionServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| isEnterprise | 判断是否企业用户 | 根据id查找用户，根据用户的类型判断是否是企业用户 |
| isCommonUser | 是否普通用户 | 根据id查找用户，根据用户的类型判断是否是普通用户 |
| create | 创建岗位 | 根据前端传递的岗位信息，在数据库中插入一条新的岗位数据 |
| deletePosition | 删除岗位 | 根据前端传来的企业id和岗位id，采用惰性删除的方法删除岗位。 |
| updatePosition | 更新岗位 | 根据前端传递的岗位信息，在数据库中更新这个岗位的数据 |
| getPositionInfo | 获取该企业创建的岗位信息 | 根据前端传来的企业id和岗位id，在数据库中找到相应岗位并返回 |
| getSpecifiedPosition | 获取指定岗位信息+用户是否申请该岗位 | 根据前端传来的用户id和岗位id，在数据库中找到相应岗位，同时根据用户id和岗位id判断该岗位是否已经申请 |
| getPositionsById | 获取企业的所有岗位 | 根据前端传来的企业id，匹配所有创建者为这个企业的岗位，并返回 |
| getRecommendedPositionsById | 获取推荐岗位 | 根据前端传来的用户id，首先找到该用户，获取用户的职位偏好，然后根据职位偏好查找符合的岗位并返回 |

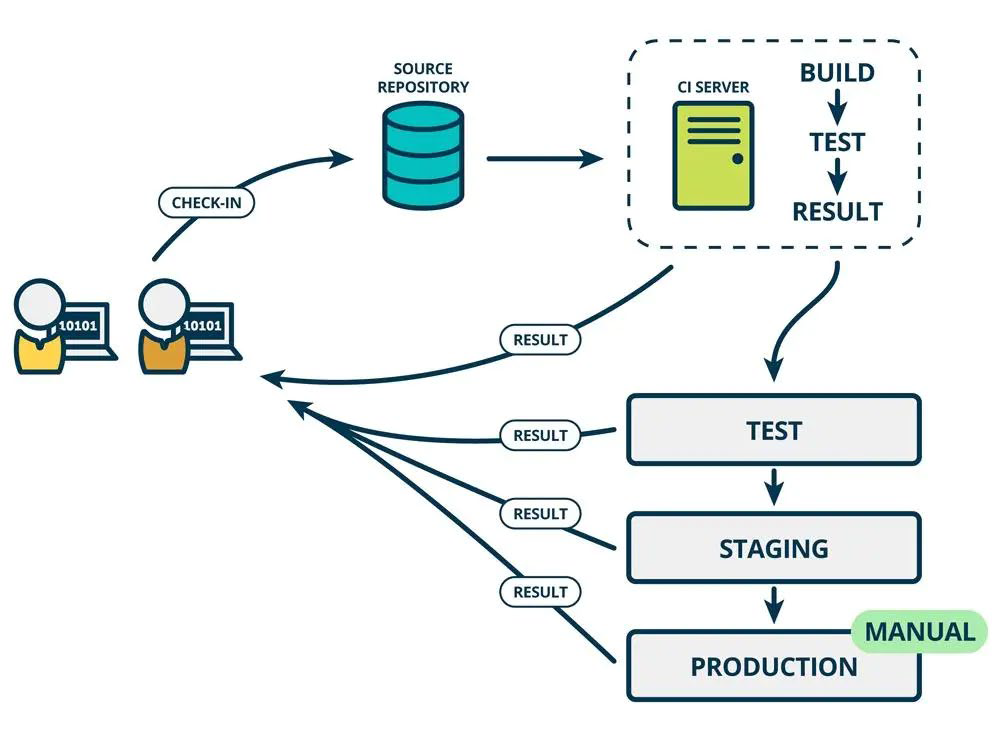
|  |  |  |
| --- | --- | --- |
| class：UserRecruitmentServiceImpl |  |  |
| 方法名 | 描述 | 实现 |
| sendApplication | 发送岗位申请 | 根据前端传来的数据在application表中插入一条新的数据 |
| cancelApplication | 取消岗位申请 | 根据前端传来的数据在application表中移除一条特定的数据 |
| getAppliedPositions | 获取申请过的岗位 | 根据前端传递用户id，匹配application表中申请者为这个id的行，在根据行中的岗位id找到岗位的详细信息并返回 |
| getAllApplicants | 企业获取某一个岗位的所有申请者 | 根据前端传来的企业id和岗位id，去application表中查找匹配的行，随后通过行中的数据分别查找用户的详细信息以及用户简历的信息并返回。 |

**5.3.5 限制条件及出错处理**

|  |  |
| --- | --- |
| class：EnterprisePositionServiceImpl |  |
| 方法名 | 描述 |
| create | 根据前端传递的岗位信息，在数据库中插入一条新的岗位数据，如果插入失败则返回0 |
| getPositionInfo | 根据前端传来的企业id和岗位id，在数据库中找到相应岗位并返回，如果没找到返回null |
| getRecommendedPositionsById | 根据前端传来的用户id，首先找到该用户，获取用户的职位偏好，然后根据职位偏好查找符合的岗位并返回，但如果某一个用户是企业用户，那么他不具有职位偏好的属性，返回空数组。 |

1. **系统部署设计**

**基于 Jenkins 的 CI/CD**



• 创建部署任务

• 连接 Github 仓库，选择分支并添加身份认证

• 配置 WebHook 钩子，添加 Jenkins 触发器

• 配置项目运行环境（后端 jdk-1.8.0，前端 node.js-16.13.1）

• 编写 **Eexcute shell** 脚本**，**指定可执行文件与日志文件

• 通过 SSH 协议连接 Linux 远程服务器

• 成功构建 Jenkins 项目，实现 CI/CD

**后端**：

• 两台 2 核 4G 的腾讯云 Linux 服务器，ip 地址分别为：101.43.31.168 和 121.5.68.218

• 各模块使用 maven 进行依赖管理与打包，开放端口如下

￮ Eureka：121.5.68.218:7001、101.43.31.168:7002

￮ User：121.5.68.218:8001、101.43.31.168:8002

￮ Tweet：121.5.68.218:8003、101.43.31.168:8004

￮ Recruitment：121.5.68.218:8005、101.43.31.168:8006

￮ Config：121.5.68.218:3344

￮ Gateway: 121.5.68.218:9527

￮ Hystrix-Dashboard: 121.5.68.218:9001

• ***Jenkins****+Github+maven* 实现项目自动化打包部署与持续交付

**前端**：

• 一台 2 核 4G 的腾讯云 Linux 服务器，ip 地址为：**121.5.68.218**，开放端口为 80

• 使用 npm 进行依赖管理与项目打包

• 同后端，采用 Jenkins 持续化部署

