|  |  |  |
| --- | --- | --- |
| **文件状态**  **[ ] 草稿**  **[ ] 讨论稿**  **[ ] 正式发布** | **文档类型** |  |
| **文件标识** |  |
| **版 本** | **1.0** |
| **作 者** |  |
| **完成日期** | **2016.3.24** |

项目名称：

山东省人力资源市场数据采集系统

文档名称：

详细设计说明书文档修订

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **更改人** | **描述（注明修改的条款或页）** |
| 1.0 | 2016.3.05 | xxx | 详细描述，尽量详细 |
| 1.0 | 2016.3.18 | 于伟平 | 编写引言部分 |
| 1.0 | 2016.3.20 | 于伟平 | 编写功能设计的控制器部分 |
| 1.0 | 2016.3.21 | 于伟平 | 编写功能设计JavaBean部分 |
| 1.0 | 2016.3.22 | 于伟平 | 编写功能设计接口部分 |
| 1.0 | 2016.3.23 | 于伟平 | 编写功能设计服务层部分 |
| 1.0 | 2016.3.24 | 于伟平 | 编写功能设计工具包部分和数据库设计部分 |

批准人签字

|  |  |  |
| --- | --- | --- |
| **职务** | **姓名** | **日期** |
| 项目经理 | 王韬懿 | 2016年3月5日 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

山东省人力资源市场数据采集系统

详细设计说明书

**目 录**

[一 引言 4](#_Toc447182581)

[1.1 编写目的和范围 4](#_Toc447182582)

[1.2 术语表 5](#_Toc447182583)

[1.3 参考资料 5](#_Toc447182584)

[1.4 使用的工具或软件 5](#_Toc447182585)

[二 功能设计 5](#_Toc447182586)

[2.1 com.data.job.controller 5](#_Toc447182587)

[CommonController 6](#_Toc447182588)

[IndexController 6](#_Toc447182589)

[PeopleController 7](#_Toc447182590)

[PostController 7](#_Toc447182591)

[ResumeController 8](#_Toc447182592)

[UserController 9](#_Toc447182593)

[2.2 com.data.job.domain 11](#_Toc447182594)

[BasicResume 11](#_Toc447182595)

[Category 11](#_Toc447182596)

[CategorySelectResults 12](#_Toc447182597)

[Education 12](#_Toc447182598)

[Employee 12](#_Toc447182599)

[Employer 12](#_Toc447182600)

[Experience 13](#_Toc447182601)

[Resume 13](#_Toc447182602)

[Skill 13](#_Toc447182603)

[2.3 com.data.job.mapper 13](#_Toc447182604)

[CategoryMapper 13](#_Toc447182605)

[EducationMapper 14](#_Toc447182606)

[EmployeeMapper 14](#_Toc447182607)

[EmployerMapper 15](#_Toc447182608)

[ExperienceMapper 15](#_Toc447182609)

[ResumeMapper 15](#_Toc447182610)

[SkillMapper 16](#_Toc447182611)

[2.4 com.data.job.service 16](#_Toc447182612)

[CategoryService 16](#_Toc447182613)

[EducationService 17](#_Toc447182614)

[EmployeeService 17](#_Toc447182615)

[EmployerService 18](#_Toc447182616)

[ExperienceService 18](#_Toc447182617)

[ResumeService 19](#_Toc447182618)

[SkillService 20](#_Toc447182619)

[XXXServiceImpl 20](#_Toc447182620)

[2.4 com.data.job.util 20](#_Toc447182621)

[BeanUtil 20](#_Toc447182622)

[Constant 20](#_Toc447182623)

[DateUtil 20](#_Toc447182624)

[FileUploader 21](#_Toc447182625)

[JsonUtil 21](#_Toc447182626)

[MD5 21](#_Toc447182627)

[Pager 21](#_Toc447182628)

[Result 21](#_Toc447182629)

[三 数据库设计 22](#_Toc447182630)

[3.1 登录信息表 22](#_Toc447182631)

[3.2 用户权限控制表 22](#_Toc447182632)

[3.3 省管理部门记录数据表 22](#_Toc447182633)

[3.4 企业数据表 23](#_Toc447182634)

[3.5 地区数据表 23](#_Toc447182635)

# 一 引言

## 1.1 编写目的和范围

在本数据采集系统系统项目的前一阶段，也就是需求分析阶段中，已经将系统用户对本系统的需求做了详细阐述。并且已经在需求分析的基础上，对数据采集系统做概要设计。

本阶段，将会在概要设计的基础上对系统进行详细设计。本阶段主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的详细设计报告中将对在本阶段中对系统所做的所有详细设计进行详细的说明。在之后的程序开发中，程序设计员可参考此详细设计报告，对系统进行详细设计,并实现相关的代码编写。在以后的软件测试阶段也可参考此说明书，以便于了解在详细设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

## 1.2 术语表

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | 客户 | 本次课程验收的老师 |
| 2 | 业务(需求)分析人员 | 项目管理者及组员 |
| 3 | 用户 | 使用本系统的省用户、市用户、企业用户等 |
| 4 | 项目 | 数据采集系统项目组 |
| 5 | 服务器 | 指本项目所采用的服务器 |
| 6 | 客户端 | 运行在用户电脑上的浏览器 |
| 9 | php | 一种用于开发服务器端的脚本语言 |
| 11 | PhpStrom | 用于开发Php的编译器 |
| 12 | Sql | 结构化查询语言 |
| 13 | Nginx | 一款高性能的web服务器 |

## 1.3 参考资料

## 1.4 使用的工具或软件

文字处理软件：word

绘图工具：processon

操作系统：Mac OSX 或 Ubuntu

IDE：IntegliU IDEA 15

数据库：Mysql5.6

数据库管理工具：Navicat Premium

版本控制工具： Git2.7.3

Jar管理 :Maven3.3

JDK: 1.7.0

Tomcat: 1.8.0

编码：UTF-8

# 二 功能设计

## 2.1 com.shandong.human.resource.controller

这是控制器包，主要包括

### CommonController

公共控制器

CommonController

静态资源前缀

private static final String STATIC\_PREFIX = "human-resource";

404错误页面

public String error\_404()

505错误页面

public String error\_500()

### ErrorController

错误页显示页面

public String error(Model model , HttpServletRequest request)

### IndexController

首页控制器

首页显示页面

public String index(Model model, HttpSession session)

### LogIOController

登入登出控制器

转登录界面

public String toLoginPage(Model model)

登录

public void login(String userName, String password, HttpSession session,Model model, HttpServletRequest request, HttpServletResponse response)

登出

public void logout(Model model, HttpSession session,HttpServletRequest request, HttpServletResponse response)

### CompanyController

公司控制器

Company

获取所有的城市

获取数据显示页面

public String getProvince(Model model)

获取所有的地区

获取数据显示页面

public ArrayList<Area> getAllArea(@PathVariable("id") Integer id)

保存企业信息

处理修改请求

public Map saveMessage(Company company,HttpSession session)

企业信息修改

获取修改页面

Public String getProvinceEdit(Model model,HttpSession session,HttpServletResponse response)

保存修改信息

public Map saveInfo(Company company,HttpSession session)

### DataController

数据控制器

添加原因和类型

public String add(Model model,HttpServletRequest request, HttpSession session,HttpServletResponse response)

保存数据

public Map onSave(CompanyData companyData,HttpSession session)

保存企业信息

获取图形显示页面

public Map<String,String> postMessage(CompanyData companyData, HttpSession session)

获取修改页面

public String query(Model model)

### AuthController

权限管理类

获取权限树并转到添加权限页面

public String toAddPage(String toPageNum, HttpServletRequest request, HttpServletResponse response)

对提交的添加权限信息进行处理

public

Result addAuth(Integer pid, String name, String url,HttpServletRequest request, HttpServletResponse response)

转到删除权限页面

public String toDeletePage(HttpServletRequest request, HttpServletResponse response)

根据ID删除指定权限及其子权限

Result deleteAuth(HttpServletRequest request, HttpServletResponse response)

从数据库中删除传入的权限树所包含的该权限树等级的权限及其子权限

private void deleteAuthTree(AuthTree tree)

### AuthRoleController

权限控制器

转到角色权限设置页面

ublic String toEditPage(Integer role\_id, Model model,HttpServletRequest request, HttpServletResponse response)

角色权限设置

public void submitHandle(Integer roleId, String authIds, HttpServletRequest request, HttpServletResponse response)

判断某个权限是否在已经存在的角色列表中

private Boolean authInAuthRole(Integer authId, List<AuthRole> existAuthRole)

### UserRoleController

用户角色控制器

转到用户角色设置页面

ublic String toEditPage(@Param("uid") Integer uid, Model model,

HttpServletRequest request, HttpServletResponse response)

用户角色设置

public void submitHandle(String[] roles, String user\_id, HttpServletRequest request, HttpServletResponse response)

判断一个roleId是否在一个角色列表中

private Boolean roleIdInUserRole(Integer roleId, List<UserRole> hasRole)

### RoleController

用户角色控制器

添加角色

public

Result add(String name, HttpServletRequest request, HttpServletResponse response)

删除角色

public void delete(String id, HttpServletRequest request, HttpServletResponse response)

显示角色

public String toRolePage(String pageNum,

HttpServletRequest request, HttpServletResponse response)

### SurveyTimeController

权限管理类

转到surveyTime页面

public String toServeyTimePage(HttpServletRequest request, HttpServletResponse response)

提交surveyTime

public void add(String date, HttpServletResponse response)

### UserController

用户控制器

用户列表

public String toShowPage(String pageNum, Model model,

HttpServletRequest request, HttpServletResponse response)

新增用户

public

Result addUser(@Valid User user, BindingResult result,

HttpServletRequest request, HttpServletResponse response)

## 2.2 com.shandong.human.resource.domain

这是本项目的JavaBean类，主要包括：

### Area

地区类

private int id; //主键

private int pid; //上级id

private String name; //地区名字

private short status; //状态

### Auth

权限类

private Integer id; // 主键

private String name; // 权限名称

private String url; // 权限对应的访问url

private Integer pid; // 上级权限

private Integer level; // 权限等级，1:一级菜单,2:二级菜单,3:三级菜单,4:页面按钮级别的权限

private Boolean hasAuth; // 判断某个角色是否有这个权限

private List<Auth> childList; // 儿子权限列表,只为遍历方便使用

### AuthRole

权限控制类

private Integer auth\_id; //权限id

private Integer role\_id; //角色id

### Company

公司类

private int id; //主键

private int province\_id; //省id

private int city\_id; //市id

private int area\_id; //区id

private String address; //地址

private String code; //组织机构代码,只可输入字母、数字，不超过9位

private String name; //企业名称

private String business; //主要经营业务

private String contact; //联系人,中文、英文

private String zipcode; //邮政编码,只可填写6位数字

private String phone; //联系电话,格式必须符合（区号）+电话号码或者为手机号码

private String fax; //传真,格式必须符合（区号）+电话

private String email; //EMAIL,格式必须符合xxx@xxx.xxx

### CompanyData

公司数据类

private int id; //主键

private int pid; //父id，默认为0，表示该数据没有被修改，若不为0，则可以通过字段查找到原始数据

private int company\_id; //company中的id

private int survey\_time\_id; //调查时间id

private int init\_people; //建档期就业人数,填写初次将档时监测点就业人数

private int cur\_people; //调查期就业人数,填写本次调查期当时的监测点就业人数

private String other\_reason; //其他原因

private String reduce\_type; //减少原因

private String reason1; //原因1

private String reason1\_explain; //原因1解释

private String reason2; //原因2

private String reason2\_explain; //原因2解释

private String reason3; //原因3

private String reason3\_explain; //原因3解释

private int status; //状态:-1:审核不通过,0:等待审核,1:审核通过

private String not\_pass\_reason; //审核不通过原因

private Date create\_time; //创建时间

private Date update\_time; //更新时间

### Log

登录类

private int id; //主键

private int user\_id; //用户id

private Date create\_time; //生成时间

private int data\_id; //修改的数据id

private String note; //备注

### MonitorInformation

系统信息类

可使用内存

private long totalMemory;

剩余内存

private long freeMemory;

最大可使用内存

private long maxMemory;

操作系统

private String osName;

总的物理内存

private long totalMemorySize;

剩余的物理内存

private long freePhysicalMemorySize;

已使用的物理内存

private long usedMemory;

线程总数

private int totalThread;

cpu使用率

private double cpuRatio;

### News

信息类

private Integer id;主键

private Integer userId;发布者ID

private String title;题目

private String content;内容

private Date createTime;创建时间

### Reduce

就业减少类

private int id; //主键

private String name; //

private int type; //类型，1:就业人数减少类型, 2:就业人数减少原因

### Role

角色类

private Integer id; //主键

private String name; //角色名

private short status; //状态,-1: 禁用，1:正常

private Boolean hasRole; //在用户授权角色页面判断是否有这个角色

### StatisticsOfCompany

公司统计类

private String city;所在城市

private Integer number;数量

private String proportion;比例

### SurveyTime

报表时间类

private int id; //主键

private Date start\_time;//开始时间

private Date end\_time; //结束时间

### User

用户类

private int id; //主键

private String username; //用户名

private String password; //MD5加密密码

private int type; //1:省用户，2:企业用户

### UserRole

用户角色类

private Integer user\_id; //用户id

private Integer role\_id; //角色id

## 2.3 com.shandong.human.resource.mapper.

### AreaMapper

地区类Mapper

获取所有的省份

public ArrayList<Area> getAllProvince();

获取所有的城市

public ArrayList<Area> getAllCity()

获取所有的县

public ArrayList<Area> getAllAreaById(int id)

获取相应的id

public Area getById(int id)

### ReduceMapper

就业减少类接口

就业减少列表

public ArrayList<Reduce> getTypeOfReason(int type);

### AuthMapper

auth表管理

auth表管理

向auth表中插入一个数据

public int insertAuth(Auth auth);

获取表中所有数据

public List<Auth> selectAll();

根据id查询数据

public Auth selectByID(int id);

根据ID删除数据

public int deleteByID(int id);

### AuthRoleMapper

角色角色类接口

插入角色权限

public int insertAuthRole(Map<Object,Object> authRole);

删除角色权限

public int deleteAuthRole(Map<Object,Object> authRole);

根据id查询角色所有权限

public List<AuthRole> selectByRoleID(int role\_id);

删除角色的所有权限

public int deleteByRoleID(int role\_id);

### CompanyMapper

公司类接口

根据id获得公司详情

public Company getCompanyById(@Param("id") Integer id);

获取所有Company

public ArrayList<Company> getAllCompany();

按条件查找Company

public ArrayList<Company> getCompanyByCondition(@Param("city\_id")String city\_id, @Param("area\_id")String area\_id, @Param("name")String name, @Param("code")String code, @Param("contact")String contact);

按条件查找Company

public Company getCompanyById(int id);

更新Company

public void updateCompanyInfo(Company company);

根据name获取CompanyID

public Company getCompanyIdByName(@Param("name")String name);

### NewsMapper

通知Mapper

新增通知

int addNews(News news);

通知列表

List<News> newsList();

删除通知

void deleteNewsById(Integer id)

查询通知

News selectNewsById(int id);

编辑通知

int editNewsById(News news);

### RoleMapper

RoleMapper

获取所有角色

public List<Role> selectAll();

插入一个角色

public int insertRole(String name);

删除一个角色

public int deleteRole(Integer id);

按照id查询角色

public int selectByID(Integer id);

### SurveyTimeMapper

调查时间接口

获取所有调查时间

public ArrayList<SurveyTime> getAllSurveyTime();

向表中插入一个时间

public Integer insertSurveyTime(SurveyTime time);

### UserMapper

用户接口

User

获取用户数

public int getCount();

查询所有用户

public List<User> selectAll();

按位置查询

public List<User> selectByPos(Map<Object , Object> pos);

插入用户

public int insertUser(User user);

删除用户

public int deleteByID(Integer id);

按用户名及密码查询用户

public User selectByNamePwd(Map<Object,Object> user);

### UserRoleMapper

UserRole接口

获取所有用户角色信息

public List<UserRole> getAll();

获取指定用户的角色信息

public List<UserRole> getRoleByUserID(Integer id);

插入以个用户角色

public int insertUserRole(Map<Object,Object> userRole);

删除一个用户角色

public int delectUserRole(Map<Object,Object> userRole);

删除一个用户的所有角色

public int delectByUserID(Integer user\_id);

## 2.4 com.data.job.service

这是项目的service层，主要包括：

### CityService

与城市相关的接口

public Integer[] getCompanyIdByCity(Integer city\_id);

public ArrayList<CompanyData> getCompanyDataByCompanyIdStatus0(Integer company\_id);

public void cityCheckPass(Integer id);

public void cityCheckFail(Integer id,String reason);

### AreaService

与地区相关接口

获取所有的省份

public ArrayList<Area> getAllProvince();

获取所有的城市

public ArrayList<Area> getAllCity();

按ID获取地区内城市

public ArrayList<Area> getAllAreaById(int id);

按ID获取地区

public Area getById(int id);

### CompanyService

与公司相关接口

保存输入信息

public void setCompanyInfo(Company company);

获取所有公司

public ArrayList<Company> isNull(int id);

按ID获取公司

public Company getCompanyById(int id);

公司公司信息

public void updateCompanyInfo(Company company);

### ReduceService

与就业减少相关的接口

获取就业人数减少类型和理由

public ArrayList<Reduce> getTypeOfReason(int type);

### AuthRoleService

与AuthRole相关的接口

AuthRole

插入角色权限

public int insertAuthRole(int auth\_id, int role\_id);

删除角色权限

public int deleteAuthRole(int auth\_id, int role\_id);

根据id查询角色所有权限

public List<AuthRole> selectByRoleID(int role\_id);

删除角色的所有权限

public int deleteByRoleID(int role\_id);

### AuthService

与Auth相关的接口

插入权限

public int insertAuth(Auth auth);

获取所有权限

public List<Auth> selectAll();

查询权限

public Auth selectByID(int id);

删除权限

public int deleteByID(int id);

### NewsService

News相关接口

News

与News相关的接口

主要包括：

1.查看通知

2.增加通知

3.删除通知

4.修改通知

新增通知

int addNews(News news);

查询所有通知

List<News> newsList();

删除通知

void deleteNewsById(Integer id);

查询通知

News selectNewsById(int id);

编辑通知

int editNewsById(News news);

### RecordService

与Record相关接口

RecordService

通过id获取Company对象

public Company getCompanyById(Integer id);

获得所有SurveyTime对象

public ArrayList<SurveyTime> getAllSurveyTime();

通过id获取指定的CompanyData对象

public ArrayList<CompanyData> getCompanyDataByCompanyId(Integer id);

获取全部的Company对象

public ArrayList<Company> getAllCompany();

通过条件查找Company对象

public ArrayList<Company> getCompanyByCondition(String city\_id, String area\_id, String name, String code, String contact);

### ReportService

与Report相关接口

获取全部CompanyData

public ArrayList<CompanyData> getAllCompanyData();

根据company\_id获取CompanyData

public ArrayList<CompanyData> getCompanyDataByCompanyId(Integer id);

根据id获取CompanyData

public CompanyData getCompanyDataById(Integer id);

根据id获取更改CompanyData的status

public void changeCompanyDataStatusById(Integer id);

按公司名获取公司

public int getCompanyIdByName(String name);

按ID获取公司信息

public ArrayList<CompanyData> getCompanyDataByCompanyIdS2(Integer id);

### RoleService

与Role相关接口

获取所有角色

public List<Role> selectAll();

插入一个角色

public int insertRole(String name);

删除一个角色

public int deleteRole(Integer id);

按照id查询角色

public int selectByID(Integer id);

### SurveyTimeService

与SurveyTime相关接口

SurveyTime

获取所有调查时间

public ArrayList<SurveyTime> getAllSurveyTime();

向表中插入一个时间

public Integer insertSurveyTime(SurveyTime time);

### UserRoleService

与UserRole相关接口

获取所有用户角色信息

public List<UserRole> getAll();

获取指定用户的角色信息

public List<UserRole> getRoleByUserID(Integer id);

插入一个用户角色

public int insertUserRole(int user\_id,int role\_id);

删除一个用户角色

public int delectUserRole(int user\_id,int role\_id);

删除一个用户的所有角色

public int delectByUserID(Integer user\_id);

### UserServiceService

与UserService相关接口

UserService

获取用户数

public int getCount();

查询所有用户

public List<User> selectAll();

按位置查询

public List<User> selectByPos(Integer offset,Integer size);

插入用户

public int insertUser(User user);

删除用户

public int deleteByID(Integer id);

按用户名及密码查询用户

public User selectByNamePwd(String username,String password);

### XXXServiceImpl

对应的各个接口的实现类，实现的方法与接口定义的一致

## 2.5 com.data.job.util

这是项目的工具包主要包括：

### BeanUtil

JavaBean工具类

主要包括：

将一个对象里面的字段进行解码处理

### Constant

常量类

### DateUtil

日期格式化工具类

主要包括

1.字符串转为日期

2.字符串转为日期

### FileUploader

文件上传工具类

主要包括：

1.实现单文件上传

2.实现多个文件的上传

### JsonUtil

Json字符串处理工具类

主要包括：

1.将对象转为JSON字符串(忽略NULL值)

2.将对象转为JSON字符串(不忽略NULL值)

3.将JSON字符串转为对象

4.从请求体中读取客户端发送的JSON串

5.回写响应

### MD5

加密

### Pager

自定义分页类

分页信息主要包括几个：

当前页|每页显示数|总页数|总数量|分页返回的信息

### Result

处理结果类

# 三 数据库设计

## 3.1 登录信息表

log：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL AUTO\_INCREMENT | 主键 |
| user\_id | int(11) unsigned NOT NULL | 用户id |
| create\_time | datetime NOT NULL | 生成时间 |
| data\_id | int(11) unsigned NOT NULL | 修改的数据id |
| note | varchar(255) DEFAULT NULL | 备注 |

## 3.2 用户权限控制表

auth：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL AUTO\_INCREMENT | 主键 |
| name | varchar(255) NOT NULL | 权限名称 |
| url | varchar(255) NOT NULL | 权限对应的访问url |
| pid | int(11) unsigned NOT NULL | 上级权限 |
| level | int(1) unsigned NOT NULL | 权限等级 |

auth\_role：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL AUTO\_INCREMENT | 主键 |
| auth\_id | int(11) unsigned NOT NULL | 权限id |
| role\_id | int(11) unsigned NOT NULL | 角色id |

## 3.3 省管理部门记录数据表

company：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL | 主键 |
| province\_id | int(11) unsigned NOT NULL | 省份id |
| area\_id | int(11) unsigned NOT NULL | 区域id |
| address | varchar(100) NOT NULL | 联系地址 |
| code | varchar(9) NOT NULL | 组织机构代码,只可输入字母、数字，不超过9位（统一编码规范） |
| name | varchar(60) NOT NULL | 企业名称 |
| business | varchar(255) NOT NULL | 主要经营业务\r是\r按实际情况填写企业主要经营的业务\r主要经营业务\r是\r按实际情况填写企业主要经营的业务\r主要经营业务,按实际情况填写企业主要经营的业务 |
| contact | varchar(20) NOT NULL | 联系人,中文、英文 |
| zipcode | varchar(6) NOT NULL | 邮政编码,只可填写6位数字 |
| phone | varchar(12) NOT NULL | 联系电话,格式必须符合（区号）+电话号码或者为手机号码 |
| fax | varchar(12) NOT NULL | 传真,格式必须符合（区号）+电话 |
| email | varchar(30) DEFAULT NULL | EMAIL,格式必须符合xxx@xxx.xxx |

## 3.4 企业数据表

company\_data：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL AUTO\_INCREMENT | 主键 |
| pid | int(11) unsigned NOT NULL DEFAULT '0' | 父id，默认为0，表示该数据没有被修改，若不为0，则可以通过字段查找到原始数据 |
| company\_id | int(11) unsigned NOT NULL |  |
| survey\_time\_id | int(11) unsigned NOT NULL | 调查期时间id |
| init\_people | int(10) unsigned NOT NULL | 建档期就业人数,填写初次将档时监测点就业人数 |
| cur\_people | int(10) unsigned NOT NULL | 调查期就业人数,填写本次调查期当时的监测点就业人数 |
| other\_reason | varchar(255) NOT NULL | 其他原因 |
| reduce\_type | varchar(50) DEFAULT NULL |  |
| reason1 | varchar(255) DEFAULT NULL | 主要原因 |
| reason1\_explain | varchar(255) DEFAULT NULL | 主要原因说明 |
| reason2 | varchar(255) DEFAULT NULL | 次要原因 |
| reason2\_explain | varchar(255) DEFAULT NULL | 次要原因说明 |
| reason3 | varchar(255) DEFAULT NULL | 第三原因 |
| reason3\_explain | varchar(255) DEFAULT NULL | 第三原因说明 |
| status | int(1) NOT NULL | 状态:-1:审核不通过,0:等待审核,1:审核通过 |
| not\_pass\_reason | varchar(255) DEFAULT NULL | 审核不通过原因 |
| create\_time | datetime NOT NULL | 创建时间 |
| update\_time | datetime NOT NULL | 更新时间 |

## 3.5 地区数据表

area：

|  |  |  |
| --- | --- | --- |
| id | int(11) unsigned NOT NULL AUTO\_INCREMENT | 主键 |
| pid | int(11) unsigned NOT NULL DEFAULT '0' | 上级id |
| name | varchar(10) NOT NULL | 地区名字 |
| status | tinyint(1) NOT NULL DEFAULT '1' | 状态 |