|  |  |  |
| --- | --- | --- |
| **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** | **2016.3.05** |

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 标题一 3](#_Toc444935381)

[1.1 小标题1 3](#_Toc444935382)

[1.2 小标题2 3](#_Toc444935383)

[二 标题二 3](#_Toc444935384)

[2.1 小标题1 3](#_Toc444935385)

[2.2 小标题2 3](#_Toc444935386)

[三 标题三 4](#_Toc444935387)

[3.1 小标题1 4](#_Toc444935388)

# 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.data.job.controller

这是控制器包，主要包括

### CommonController

公共控制器

包括用户的登录、注册、退出登录等

静态资源前缀

public static final String STATIC\_PREFIX = "job";

private EmployeeService employeeService;

private EmployerService employerService;

登录页面

public String showLogin(Model model)

退出登录

public String logout

注册Employee

Integer id = employerService.addEmployee(employee);

注册Employer

Integer id = employerService.addEmployer(employer);

Employee登录

Employee loginEmployee = employeeService.loginEmployee(employee);

Employer登录

Employer loginEmployer = employerService.loginEmployer(employer);

### IndexController

首页控制器

包括首页显示、404、500页面

静态资源前缀

public static final String STATIC\_PREFIX = "job";

自动加载 EmployeeService

private EmployeeService employeeService;

自动加载 ResumeService

private ResumeService resumeService;

首页显示页面

public String index(Model model)

404页面配置

public String fourZero()

500页面配置

public String fiveZero()

### PeopleController

人才页面控制器

包括：1.人才列表 2.加载人才列表

静态资源前缀

public static final String STATIC\_PREFIX = "job/people";

自动加载 EmployeeService

private EmployeeService employeeService;

人才列表

public String people(HttpSession session, Model model)

加载人才列表

public String loadPeople(Model model, String param, Pager<Employee> pager)

### PostController

发布简历控制器

主要包含以下功能:

1.发布简历基本信息

2.新增、编辑简历基本信息

3.发布、编辑简历详情

4.增加、编辑、删除教育信息

5.增加、编辑、删除工作经历

6.新增、编辑、删除技能

7.预览简历

静态资源前缀

public static final String STATIC\_PREFIX = "job/post";

自动加载CategoryService

private CategoryService categoryService;

自动加载CityService

private CityService cityService;

自动加载ResumeService

private ResumeService resumeService;

自动加载EducationService

private EducationService educationService;

自动加载ExperienceService

private ExperienceService experienceService;

自动加载SkillService

private SkillService skillService;

检查是否可以发布简历

Result postResume(HttpSession session)

发布简历基本信息

public String postInfo(Model model, HttpSession session)

编辑简历基本信息

public String postEditInfo(Model model, @PathVariable Integer id)

新增简历基本信息

Result handlePostInfo(Resume resume, HttpSession session)

编辑简历基本信息

Result handleEditPostInfo(Resume resume, HttpSession session, @PathVariable("id") Integer resumeId)

编辑简历详情

public String postEditDetail(@PathVariable("id") Integer id, Model model)

发布简历详情

public String postDetail() throws IOException {

增加教育信息

Result addEducation(Education education, HttpSession session)

编辑教育信息

Result editEducation(Education education, HttpSession session)

增加工作经历

Result addExperience(Experience experience, HttpSession session)

编辑工作经历

Result editExperience(Experience experience, HttpSession session)

新增技能

Result addSkill(Skill skill, HttpSession session)

编辑技能

Result editSkill(Skill skill, HttpSession session)

预览简历

public String postPreview(Model model, HttpSession session)

预览简历详情

public String postPreviewDetail(@PathVariable("id") Integer id, Model model, HttpSession session)

删除教育信息

Result deleteEdu(String id)

删除工作经验

Result deleteExp(String id)

删除技能

Result deleteSkill(String id)

### ResumeController

简历控制器

包括：

1.简历列表

2.加载简历列表

3.简历详情

静态资源前缀

public static final String STATIC\_PREFIX = "job/resume";

自动加载EmployeeService

private EmployeeService employeeService;

自动加载CategoryService

private CategoryService categoryService;

自动加载CityService

private CityService cityService;

自动加载 EmployeeService

private ResumeService resumeService;

自动加载EducationService

private EducationService educationService;

自动加载ExperienceService

private ExperienceService experienceService;

自动加载SkillService

private SkillService skillService;

简历列表

public String resume(HttpSession session, Model model)

加载简历列表

public String loadResume(Model model, String param, Pager<Resume> pager)

某一个简历详情

public String resumeDetail(@PathVariable("id") Integer id, Model model)

### UserController

网站用户的管理控制器

该控制器下面的所有操作均需要以后才能使用

包括以下功能:

1.我的账户页面

2.账户管理页面

3.我的简历列表

4.某一个简历详情

5.上传用户头像

6.更新用户资料

7.修改密码

8.根据简历id删除简历

9.更新简历的状态

静态资源前缀

public static final String STATIC\_PREFIX = "job/user"

自动加载EmployeeService

private EmployeeService employeeService;

自动加载ResumeService

private ResumeService resumeService;

自动加载CategoryService

private CategoryService categoryService;

自动加载CityService

private CityService cityService;

自动加载EducationService

private EducationService educationService;

自动加载ExperienceService

private ExperienceService experienceService;

自动加载SkillService

private SkillService skillService

我的账户页面

public String index(Model model, HttpSession session)

账户管理页面

public String account(Model model, HttpSession session)

我的简历列表

public String resumeList(Model model, HttpSession session, HttpServletRequest request)

某一个简历详情

public String resumeDetail(@PathVariable("id") Integer id, Model

model)

网站用户的管理控制器

该控制器下面的所有操作均需要以后才能使用

包括以下功能:

1.我的账户页面

2.账户管理页面

3.我的简历列表

4.某一个简历详情

5.上传用户头像

6.更新用户资料

7.修改密码

8.根据简历id删除简历

9.更新简历的状态

静态资源前缀

public static final String STATIC\_PREFIX = "job/user"

自动加载EmployeeService

private EmployeeService employeeService;

自动加载ResumeService

private ResumeService resumeService;

自动加载CategoryService

private CategoryService categoryService;

自动加载CityService

private CityService cityService;

自动加载EducationService

private EducationService educationService;

自动加载ExperienceService

private ExperienceService experienceService;

自动加载SkillService

private SkillService skillService

我的账户页面

public String index(Model model, HttpSession session)

账户管理页面

public String account(Model model, HttpSession session)

我的简历列表

public String resumeList(Model model, HttpSession session, HttpServletRequest request)

某一个简历详情

public String resumeDetail(@PathVariable("id") Integer id, Model

model) {

上传用户头像

Result uploadAvatar(@RequestParam("file") MultipartFile file, HttpSession session,HttpServletRequest request, HttpServletResponse response)

更新用户资料

Result updateProfile(HttpSession session, HttpServletRequest request)

修改密码

Result changePwd(HttpSession session, HttpServletRequest request)

根据简历id删除简历

Result deleteResume(String id)

更新简历的状态

Result updateResumeStatus(Integer id, Integer status, HttpSession session)

## 2.2 com.data.job.domain

这是本项目的JavaBean类，主要包括：

### BasicResume

基本简历信息实体类

private Resume resume;//数据库对应resume表

private List<Category> categoryList;//简历中的工作分类

private List<City> cityList;//城市列表

### Category

工作种类

private Integer id;//主键

private String name;//名称

private String title;//标题

### CategorySelectResults

类别选择结果

private Category category;

private boolean chosen;

### Education

简历上面的教育信息

private Integer id;//主键

private String school;//学校

private String qualification;//资质

private String date;//开始截止时间

private String note;//备注

private Integer resumeId;//简历的id

### Employee

Employee 模型

private Integer id;//id

private String username;//用户名

private String password;//密码

private String email;//邮箱

private String job;//当前工作

private Date birthday;//生日

private String address;//地址

private String phone;//手机号

private String description;//介绍

private String avatar;//头像

### Employer

找人者Model

private Integer id;//id

private String username;//用户名

private String password;//密码

private String email;//邮箱

### Experience

工作经历Model

private Integer id;//主键

private String title;//称呼

private String date;//开始截止时间

private String note;//备注

private Integer resumeId;//简历id

### Resume

简历模型

private Integer id;//主键

private String title;//标题

private Integer employeeId;//人才id

private String highestDegree;//最高学历

private String experience;//工作经历年限

private String introduction;//自我介绍

private String jobCategoryIds;//工作种类id

private String cityIds;//所在城市id

private Integer status;//是否可见,1=>可见,0=>不可见

### Skill

Skill 模型

private Integer id;//主键

private String name;//技能名称

private Integer percent;//百分比

private Integer resumeId;//简历id

## 2.3 com.data.job.mapper

### CategoryMapper

工作种类Mapper

获得种类列表

List<Category> getCategoryList();

根据id获得工作种类

Category getCategoryById(@Param("id") Integer id);

### EducationMapper

EducationMapper

新增教育信息

Education getSpecificEducation(@Param("resumeId") Integer resumeId, @Param("school") String school, @Param("qualification") String qualification, @Param("date") String date);

获得教育列表

List<Education> getEducationList(@Param("resumeId") Integer resumeId);

根据id删除教育信息

int deleteEducationById(@Param("id") Integer id);

更新教育信息

int updateEducation(Education education);

### EmployeeMapper

应聘者Mapper

通过id得到Employee

Employee getEmployeeById(@Param("id") Integer id);

新增Employee

int addEmployee(Employee employee);

根据用户名查找Employee

Employee getEmployeeByUsername(String username);

Employee 登录

Employee loginEmployee(Employee employee);

更新头像

int updateAvatar(@Param("id") Integer id, @Param("avatar") String avatar);

更新资料

int updateProfile(Employee employee);

更新密码

int updatePassword(@Param("id") Integer id, @Param("password") String password);

获得人员数量

int getEmployeeCount(HashMap<String, Object> paramMap);

获得人员列表

List<Employee> getEmployeeList(HashMap<String, Object> paramMap);

获得最新的指定数量的employee列表

List<Employee> getAEmployeeList(@Param("limit") int limit);

### EmployerMapper

招聘者接口

新增Employer

int addEmployer(Employer employer);

根据用户名查找Employer

Employer getEmployerByUsername(String username);

Employer 登录

Employer loginEmployer(Employer employer);

### ExperienceMapper

ExperienceMapper

增加工作经历

int addExperience(Experience experience);

获得特定的工作经验

Experience getSpecificExperience(@Param("title") String title, @Param("date") String date, @Param("note") String note, @Param("resumeId") Integer resumeId);

获得经历列表

List<Experience> getExperienceList(@Param("resumeId") Integer resumeId);

根据id删除工作经历

int deleteExperienceById(@Param("id") Integer id);

更新经历

int updateExperience(Experience experience);

### ResumeMapper

ResumeMapper

新增Resume

int addResume(Resume resume);

根据title查找Resume

Resume getResumeByTitle(@Param("title") String title, @Param("id") Integer id);

根据id获得简历详情

Resume getResumeById(@Param("id") Integer id);

根据用户id获得简历列表

List<Resume> getResumeListByUserId(@Param("id") Integer id, @Param("status") Integer status);

根据id删除简历

int deleteResumeById(@Param("id") Integer id);

更新resume的状态

int updateResumeStatus(@Param("id") Integer id, @Param("status") Integer status);

更新Resume

int updateResume(Resume resume);

获得简历数量

int getResumeCount(HashMap<String, Object> paramMap);

获得简历列表

List<Resume> getResumeList(HashMap<String, Object> paramMap);

获得指定数量的简历列表

List<Resume> getAResumeList(@Param("limit") int limit);

### SkillMapper

SkillMapper

新增技能

int addSkill(Skill skill);

获得某个特定的技能

Skill getSpecificSkill(@Param("name") String name, @Param("percent") Integer percent, @Param("resumeId") Integer resumeId);

获得技能列表

List<Skill> getSkillList(@Param("resumeId") Integer resumeId);

根据id删除skill

int deleteSkillById(@Param("id") Integer id);

更新技能

int updateSkill(Skill skill);

## 2.4 com.data.job.service

这是项目的service层，主要包括：

### CategoryService

种类Service

主要包括：

1.获得种类列表

2.根据id获得工作种类

获得种类列表

List<Category> getCategoryList();

根据id获得工作种类

Category getCategoryById(Integer id);

### EducationService

与教育信息相关的接口

主要包括：

1.新增教育信息

2.根据学校查找教育信息

3.找到特定的教育信息

4.获得教育列表

5.根据id删除教育信息

6.更新教育信息

新增教育信息

int addEducation(Education education);

根据学校查找教育信息

List<Education> getEducationBySchool(Integer resumeId, String school);

找到特定的教育信息

Education getSpecificEducation(Integer resumeId, String school, String qualification, String date);

获得教育列表

List<Education> getEducationList(Integer resumeId);

根据id删除教育信息

int deleteEducationById(Integer id);

更新教育信息

int updateEducation(Education education);

### EmployeeService

与Employee 相关的接口

主要包括：

1.新增Employee

2.根据用户名查找Employee

3.Employee 登录

4.通过id得到Employee

5.更新头像

6.更新资料

7.更新密码

8.获得人员列表

9.获得最新的指定数量的employee列表

新增Employee

int addEmployee(Employee employee);

根据用户名查找Employee

Employee getEmployeeByUsername(String username);

Employee 登录

Employee loginEmployee(Employee employee);

通过id得到Employee

Employee getEmployeeById(Integer id);

更新头像

int updateAvatar(Integer id, String avatar);

更新资料

int updateProfile(Employee employee);

更新密码

int updatePassword(Integer id, String password);

获得人员列表

List<Employee> getEmployeeList(HashMap<String, Object> paramMap, Pager<Employee> pager);

获得最新的指定数量的employee列表

List<Employee> getAEmployeeList(int limit);

### EmployerService

与Employer相关的接口

主要包括：

1.新增Employer

2.根据用户名查找Employer

3.Employer 登录

新增Employer

int addEmployer(Employer employer);

根据用户名查找Employer

Employer getEmployerByUsername(String username);

Employer 登录

Employer loginEmployer(Employer employer);

### ExperienceService

ExperienceService

主要包括：

1.增加工作经历

2.获得特定的工作经验

3.获得经历列表

4.根据id删除工作经历

5.更新经历

增加工作经历

int addExperience(Experience experience);

获得特定的工作经验

Experience getSpecificExperience(String title, String date, String note, Integer resumeId);

获得经历列表

List<Experience> getExperienceList(Integer resumeId);

根据id删除工作经历

int deleteExperienceById(Integer id);

更新经历

int updateExperience(Experience experience);

### ResumeService

ResumeService相关接口

主要包括：

1.新增Resume

2.根据title查找Resume

3.根据id获得简历详情

4.根据用户id获得简历列表

5.根据id删除简历

6.更新resume的状态

7.更新Resume

8.获得简历列表

9.获得指定数量的简历列表

新增Resume

int addResume(Resume resume);

根据title查找Resume

Resume getResumeByTitle(String title, Integer id);

根据id获得简历详情

Resume getResumeById(Integer id);

根据用户id获得简历列表

List<Resume> getResumeListByUserId(Integer id, Integer status);

根据id删除简历

int deleteResumeById(Integer id);

更新resume的状态

int updateResumeStatus(Integer id, Integer status);

更新Resume

int updateResume(Resume resume);

获得简历列表

List<Resume> getResumeList(HashMap<String, Object> paramMap, Pager<Resume> pager);

获得指定数量的简历列表

List<Resume> getAResumeList(int limit);

### SkillService

与SkillService相关接口

主要包括：

1.新增技能

2.获得某个特定的技能

3.获得技能列表

4.根据id删除skill

5.更新技能

新增技能

int addSkill(Skill skill);

获得某个特定的技能

Skill getSpecificSkill(String name, Integer percent, Integer resumeId);

获得技能列表

List<Skill> getSkillList(Integer resumeId);

根据id删除skill

int deleteSkillById(Integer id);

更新技能

int updateSkill(Skill skill);

### 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

## 2.4 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 |
| city\_id | int(11) unsigned NOT NULL | 城市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 |