数据库设计

目录

[数据库设计 1](#_Toc518220057)

[一、系统需求分析报告 3](#_Toc518220058)

[1.1、需求分析 3](#_Toc518220059)

[1.2、数据流图 3](#_Toc518220060)

[1.3、数据字典 4](#_Toc518220061)

[二、数据逻辑结构设计 16](#_Toc518220062)

[2.1、E-R图 16](#_Toc518220063)

[2.2 关系模式 17](#_Toc518220064)

[2.3 数据库结构 17](#_Toc518220065)

[三、数据库实现 19](#_Toc518220066)

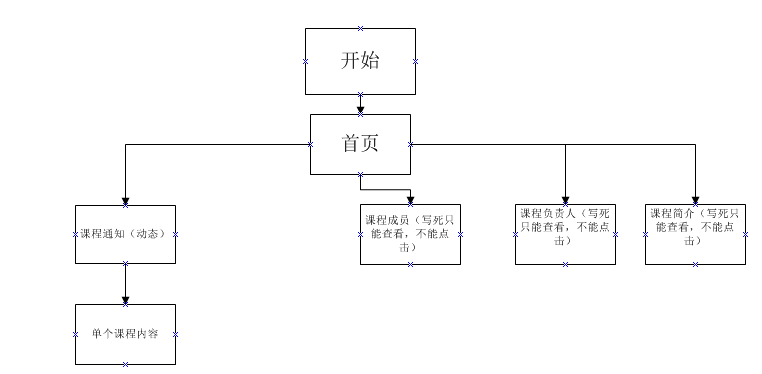
## 一、系统需求分析报告

### 1.1、需求分析

见需求分析报告

### 1.2、数据流图

1.2-1- 后台管理数据流图



### 1.3、数据字典

经过分析可以得到以下数据流条目，并且转换成后台接口文档

#### 1.3.1前台页面

1. getColumns 获取栏目信息 已完成

**前台参数**

* + level: （0表示父栏目，1表示子栏目，一共两层目录）
  + columnId: 栏目id（可选）

**后台返回 JSON数组**

* + columnId: 栏目id
  + columnName: 栏目名称
  + level: 栏目级别
  + parentId: 父栏目id

**使用**

* + 进入首页直接请level为0的栏目
  + 点击某个栏目传level为1和点击栏目的id

1. getResources 获取资源 包括文章和文件等资源(**后台管理也可以使用**) 已完成

**前台参数**

* + columnId: 栏目id
  + title: 资源标题
  + startTime: 创建时间的范围开始
  + endTime: 创建时间的范围结束
  + pageIndex: 当前页
  + pageSize: 每页接收的大小

**后台返回JSON数组**

* + resourceId: 资源id
  + title: 资源标题
  + createTime: 创建时间
  + whetherTop: 是否置顶
  + status: 资源状态
  + columnId: 栏目id
  + columnName: 栏目名称

1. getResourceContent 获取资源详情（既可以获取文章也可以获取文件）(**后台管理也可以使用**) 已完成

**前台参数**

* + resourceId: 资源id

**后台返回JSON字符串**

* + resourceId: 资源id
  + content: 资源内容
  + path: 资源路径
  + title: 资源标题
  + createTime: 创建时间
  + whetherTop: 是否置顶
  + status: 资源状态
  + columnId: 栏目id
  + columnName: 栏目名称
  + fileName: 文件名

1. search 全站搜索 已完成

**前台参数**

* + keyWords: 关键字

**后台返回JSON数组**

* + resourceId: 资源id
  + title: 资源标题
  + createTime: 创建时间
  + whetherTop: 是否置顶
  + employeeId: 作者
  + status: 文章状态
  + path: 资源路径

1. addMessage 新增留言 已完成

**前台参数**

* + content: 留言内容
  + createTime: 创建时间
  + moduleId: 板块id

**后台返回int**

* + result: 留言成功1，留言失败0

1. getMessages 获取留言(**后台管理也可以使用**) 已完成

**前台参数**

* + pageIndex: 当前页码
  + pageSize: 每页多少个
  + status 默认为1 (1为已经回复，0为未回复，2为全选，可选)

**后台返回JSON数组**

* + 当前台参数status为 1 时
    - messageId: 留言id
    - content: 留言内容
    - createTime: 留言时间
    - reply: 回复内容
    - replyTime: 回复时间
    - moduleId: 板块id
  + 当前台参数status为 0 时
    - messageId: 留言id
    - content: 留言内容
    - createTime: 留言时间
    - moduleId: 板块id
  + 当前台参数status为 2 时
    - messageId: 留言id
    - content: 留言内容
    - createTime: 留言时间
    - reply: 回复内容
    - replyTime: 回复时间
    - moduleId: 板块id
    - status: 留言状态

#### 1.3.2后台登录

1. login 登录方法 已完成

**前台参数:**

* + account
  + password

**后台返回int**

* + 1 登录成功
  + 0 密码错误
  + -1 没有这个账号

**后台设置session:**

* + account 记录用户登录的账号

**具体内容：** 通过account去数据库里查找相应的password，并且通过这个account寻找到其对应的角色roleId，再通过roleId获取相应的权限

#### 1.3.3后台管理

#### 1.3.3.1后台管理整体页面

1. getModule 获取模块 已完成

**前台参数:**

* + 无

**后台返回一个JSON数组，包含父模块基本信息和子模块基本信息**

[

moduleId: 0,

moduleName: "",

status: 0,

submodules:[

{

subModuleId: 0,

subModuleName: "",

status: 0,

parentModuleId: 0,

}

...

]

...

]

**后台实现**：通过获取session中的account账号找到对应的roleId，然后通过roleId获取所对应的权限对应的板块

#### 1.3.1网站内容管理

##### 1.3.1.1留言管理

1. deleteMessages 删除留言 已完成

**前台参数**

* + messageIds: 需要删除留言的id

格式为:

messageIds:

[

{

messageId: 0,

}

{

messageId: 1,

}

...

]

**后台返回int**

* + result: 成功1，失败0

1. replyMessage 回复留言 已完成

**前台参数** -- messageId: 留言id

* + reply: 回答内容
  + replyTime: 回答时间
  + status: 留言状态

**后台返回int**

* + result: 成功1，失败0

#### 1.3.4.2文章管理和审核文章

1. addResource 添加资源 已完成

**前台参数**

* + title: 资源名称
  + employeeId: 上传资源的人的id
  + columnId: 栏目id
  + content: 资源内容
  + createTime: 创建时间
  + whetherTop: 是否置顶

**后台返回int**

* + result: 成功1，失败0

1. addFile 上传文件

**前台参数**

* + fileName: 文件名
  + file: 文件资源

**后台返回int**

* + result: 成功1，失败0

1. updateResource 更新资源

**前台参数**

* + type:

**0 修改**

* + - resourceId: 资源id
    - title: 资源名称
    - employeeId: 更改人id
    - columnId: 栏目id
    - content: 内容
    - whetherTop: 是否置顶(0:不置顶 1:置顶)

**1 审核**

* + - status: 文章状态(0:未审核 1: 审核通过 2:审核不通过)
    - resourceId: 资源id

**后台返回int**

* + result: 成功1，失败0

1. deleteResources 删除资源 已完成

**前台参数**

* + resourceIds: 所选要删除资源的id

**后台返回int**

* + result: 成功1，失败0

#### 1.3.4.3栏目管理

1. updateColumn 更新栏目 已完成

**前台参数**

* + columnId: 栏目id
  + parentId: 父栏目id
  + columnName: 栏目名称

**后台返回int**

* + result: 成功1，失败0

#### 1.3.2系统设置

##### 1.3.2.1模块管理

1. updateModule 更新模块 已完成

**前台参数**

* + moduleId: 模块id
  + moduleName: 模块名字
  + status: 是否被禁用(0:禁用 1:启用)
  + parentModuleId: 父模块id

**后台返回int**

* + result: 成功1，失败0

##### 1.3.2.2角色分配

1. assignRole 分配角色(一个用户只能分配一个角色) 已完成

**前台参数**

* + employeeId: 被分配角色人员id
  + roleId: 分配的角色id

**后台返回int**

* + result: 成功1，失败0

##### 1.3.2.3人员管理

1. addEmployee 新增用户 已完成

**前台参数**

* + employeeName: 用户名字
  + sex: 用户性别
  + password: 密码

**后台返回int**

* + result: 成功1，失败0

1. deleteEmployees 删除用户

**前台参数**

* + employeeIds: 被删除用户的id(一个数组)

**后台返回int**

* + result: 成功1，失败0

1. getEmployees 获取人员

**前台参数**

* + roleId: (0 只有角色id为0的才是没有被分配角色的， 可选)
  + pageIndex: 当前页 (可选)
  + pageSize: 页面大小 (可选)
  + condition: 查询条件 (可选)

**后台返回JSON数组**

* + employeeId: 人员id
  + employeeName: 人员名称
  + sex: 人员性别
  + account: 人员的账号

##### 1.5.2.4角色管理

1. addRole 新增角色

**前台参数**

* + roleName: 角色名字
  + createTime: 创建时间
  + description: 角色描述
  + ownModuleIds: 角色拥有的模块id(一个数组)

**后台返回int**

* + result: 成功1，失败0

1. getRoles 获取角色

**前台参数**

* + 无

**后台返回JSON数组**

* + roleId: 角色id
  + roleName: 角色名称
  + createTime: 角色被创建时间
  + description: 角色的描述

1. updateRole 更新角色

**前台参数**

* + roleId: 角色id
  + roleName: 角色名字
  + createTime: 创建时间
  + description: 角色描述
  + ownModuleIds: 角色拥有的模块id(一个数组)

**后台返回int**

* + result: 成功1，失败0

1. deleteRoles 删除角色

**前台参数**

* + roleIds: 被删除角色的id

**后台返回int**

* + result: 成功1，失败0

1. getRoleByAccount 获取登录的单个角色信息 已完成

**前台参数**

* + 无

**后台返回JSON**

* + roleId: 角色id
  + roleName: 角色名称
  + createTime: 角色被创建时间
  + description: 角色的描述

## 二、数据逻辑结构设计

### 2.1、E-R图

### 2.2 关系模式

栏目表（栏目编号，栏目名称，栏目级别，栏目的父栏目编号）；

文章表（文章编号，文章标题， 文章作者编号， 文章内容， 文章所属栏目编号， 文章创建时间， 是否置顶， 文章状态）；

角色（角色编号， 角色名称， 角色创建时间， 角色的描述）；

权限（权限编号， 角色编号， 模块编号）；

父模块（模块编号， 模块名称， 模块状态）；

子模块（模块编号， 模块名称， 模块状态， 父模块编号）；

人员（人员编号， 人员姓名， 人员性别， 人员账号， 人员密码， 角色编号）；

留言（留言编号， 留言内容， 创建时间， 回复内容， 回复时间， 回复人员编号， 回复状态， 留言模块编号）；

教育资源（资源编号， 所属文章编号， 存放路径， 文件名称）

### 2.3 数据库结构

栏目表 colunm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 栏目编号 | int(11) | Primary key |  |
| name | 栏目名称 | Varchar(40) |  |  |
| level | 栏目级别 | smallint | 默认为0 |  |
| parentId | 栏目的父栏目编号 | Int(11) | 没有则为0，默认为0 | Colunm表外键 |

文章表 article

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 文章编号 | int(11) | Primary key |  |
| Title | 文章标题 | Varchar(40) |  |  |
| EmployeeId | 文章作者编号 | smallint | 默认为0 | Employee表外键 |
| columnId | 栏目编号 | Int(11) | 默认为0 | Colunm表外键 |
| content | 文章内容 | Varchar（255） |  |  |
| createTime | 创建时间 | dateTime |  |  |
| whethterTop | 是否置顶 | Smallint | 默认为0，(0:不置顶 1:置顶) |  |
| Status | 文章状态 | smallint | 0:未审核 1: 审核通过 2:审核不通过 |  |

角色表 role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 角色编号 | int(11) | Primary key |  |
| Name | 角色名称 | Varchar(40) |  |  |
| createTime | 创建时间 | dateTime |  |  |
| Description | 描述角色 | Varchar(40) |  |  |

权限表 permission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 权限编号 | int(11) | Primary key |  |
| roleId | 角色名称 | int(11) |  | 角色表id外键 |
| moduleId | 创建时间 | int(11) |  | 子模块表id外键 |

父模块表 module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 模块编号 | int(11) | Primary key |  |
| name | 模块名称 | Varchar(40) |  |  |
| Status | 模块状态 | Smallint | 0:禁用 1:启用 |  |

子模块表 subModule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 模块编号 | int(11) | Primary key |  |
| name | 模块名称 | Varchar(40) |  |  |
| Status | 模块状态 | Smallint | 0:禁用 1:启用 |  |
| parentModuleId | 父类模块编号 | int(11) |  | 父模块id外键 |

人员表 employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 员工编号 | int(11) | Primary key |  |
| name | 员工名称 | Varchar(40) |  |  |
| sex | 员工性别 | Smallint | (0:未知 1男性 2：女性) |  |
| account | 员工账号 | int(11) |  | 等于id |
| passWord | 员工密码 | Varchar(40) |  |  |
| RoleId | 角色编号 | int(11) |  | 角色表id外键 |

留言表 message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 留言编号 | int(11) | Primary key |  |
| Content | 留言内容 | Varchar(255) |  |  |
| createTime | 留言时间 | Datetime |  |  |
| reply | 回复内容 | Varchar(255) |  |  |
| replyTime | 留言回复时间 | Datetime |  |  |
| employeeId | 回复人编号 | int(11) |  | 人员表id外键 |
| Status | 回复状态 | smallint | (0:未回复 1:已回复) |  |
| moduleId | 留言板块编号 | Int（11） |  | 模块表id外键 |

资源表 file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 注释 | 数据类型 | 属性限定 | 关系 |
| Id | 资源编号 | int(11) | Primary key |  |
| articleId | 与文章管理相关联的编号 | int(11) |  | 文章id外键 |
| name | 文件名称 | Varchar(40) |  |  |
| Path | 存放的路径 | Varchar(255) |  |  |

## 三、数据库实现

*/\**

*Navicat MySQL Data Transfer*

*Source Server : zxh*

*Source Server Version : 80011*

*Source Host : localhost:3306*

*Source Database : general-course*

*Target Server Type : MYSQL*

*Target Server Version : 80011*

*File Encoding : 65001*

*Date: 2018-06-30 11:19:06*

*\*/*

SET FOREIGN\_KEY\_CHECKS=0;

*-- ----------------------------*

*-- Table structure for article*

*-- ----------------------------*

DROP TABLE IF EXISTS `article`;

CREATE TABLE `article` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`title` varchar(40) DEFAULT NULL,

`employeeId` int(11) DEFAULT NULL,

`columnId` int(11) DEFAULT NULL,

`content` varchar(255) DEFAULT NULL,

`createTime` datetime DEFAULT NULL,

`whetherTop` smallint(2) DEFAULT '0',

`status` smallint(2) DEFAULT '0',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100002 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of article*

*-- ----------------------------*

INSERT INTO `article` VALUES ('100000', 'helloworld', '100000', '100004', 'nihao中国', '2018-06-29 19:23:42', '1', '0');

INSERT INTO `article` VALUES ('100001', 'helloworld', '100000', '100004', 'nihao中国', '2018-06-29 19:23:42', '1', '0');

*-- ----------------------------*

*-- Table structure for colunm*

*-- ----------------------------*

DROP TABLE IF EXISTS `colunm`;

CREATE TABLE `colunm` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(40) DEFAULT NULL,

`level` smallint(2) DEFAULT '0',

`parentId` int(11) DEFAULT '0',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100012 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of colunm*

*-- ----------------------------*

INSERT INTO `colunm` VALUES ('100000', '首页', '0', '0');

INSERT INTO `colunm` VALUES ('100001', '课程介绍', '0', '0');

INSERT INTO `colunm` VALUES ('100002', '教学团队', '0', '0');

INSERT INTO `colunm` VALUES ('100003', '课程建设', '0', '0');

INSERT INTO `colunm` VALUES ('100004', '课程资源', '0', '0');

INSERT INTO `colunm` VALUES ('100005', '问题与答疑', '0', '0');

INSERT INTO `colunm` VALUES ('100006', '课程研究', '0', '0');

INSERT INTO `colunm` VALUES ('100007', '课程简介', '1', '100001');

INSERT INTO `colunm` VALUES ('100008', '教学方法', '1', '100001');

INSERT INTO `colunm` VALUES ('100009', '课程动态', '1', '100001');

INSERT INTO `colunm` VALUES ('100010', '教学课件', '1', '100004');

INSERT INTO `colunm` VALUES ('100011', '教学视频', '1', '100004');

*-- ----------------------------*

*-- Table structure for employee*

*-- ----------------------------*

DROP TABLE IF EXISTS `employee`;

CREATE TABLE `employee` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(40) DEFAULT NULL,

`sex` smallint(2) DEFAULT NULL,

`account` int(11) DEFAULT NULL,

`password` varchar(40) DEFAULT NULL,

`roleId` int(11) DEFAULT '0',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100001 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of employee*

*-- ----------------------------*

INSERT INTO `employee` VALUES ('100000', 'admin', '0', '100000', '123456', '100000');

*-- ----------------------------*

*-- Table structure for file*

*-- ----------------------------*

DROP TABLE IF EXISTS `file`;

CREATE TABLE `file` (

`name` varchar(40) DEFAULT NULL,

`id` int(11) NOT NULL AUTO\_INCREMENT,

`articleId` int(11) DEFAULT NULL,

`path` varchar(40) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100000 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of file*

*-- ----------------------------*

*-- ----------------------------*

*-- Table structure for message*

*-- ----------------------------*

DROP TABLE IF EXISTS `message`;

CREATE TABLE `message` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`content` varchar(255) DEFAULT NULL,

`createTime` datetime DEFAULT NULL,

`reply` varchar(255) DEFAULT NULL,

`employeeId` int(11) DEFAULT NULL,

`replyTime` datetime DEFAULT NULL,

`status` smallint(2) DEFAULT NULL,

`moduleId` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100000 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of message*

*-- ----------------------------*

*-- ----------------------------*

*-- Table structure for module*

*-- ----------------------------*

DROP TABLE IF EXISTS `module`;

CREATE TABLE `module` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(40) DEFAULT NULL,

`status` smallint(2) DEFAULT '1',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100002 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of module*

*-- ----------------------------*

INSERT INTO `module` VALUES ('100000', '网站内容管理', '1');

INSERT INTO `module` VALUES ('100001', '系统设置', '1');

*-- ----------------------------*

*-- Table structure for permission*

*-- ----------------------------*

DROP TABLE IF EXISTS `permission`;

CREATE TABLE `permission` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`roleId` int(11) DEFAULT NULL,

`moduleId` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100009 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of permission*

*-- ----------------------------*

INSERT INTO `permission` VALUES ('100000', '100000', '100000');

INSERT INTO `permission` VALUES ('100001', '100000', '100001');

INSERT INTO `permission` VALUES ('100002', '100000', '100002');

INSERT INTO `permission` VALUES ('100003', '100000', '100003');

INSERT INTO `permission` VALUES ('100004', '100000', '100004');

INSERT INTO `permission` VALUES ('100005', '100000', '100005');

INSERT INTO `permission` VALUES ('100006', '100000', '100006');

INSERT INTO `permission` VALUES ('100007', '100000', '100007');

INSERT INTO `permission` VALUES ('100008', '100000', '100008');

*-- ----------------------------*

*-- Table structure for role*

*-- ----------------------------*

DROP TABLE IF EXISTS `role`;

CREATE TABLE `role` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(40) DEFAULT NULL,

`createTime` datetime DEFAULT NULL,

`description` varchar(40) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100001 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of role*

*-- ----------------------------*

INSERT INTO `role` VALUES ('100000', '超级管理员', '2018-06-27 22:23:06', '所有权限都有的人，相当于Linux下的root用户');

*-- ----------------------------*

*-- Table structure for submodule*

*-- ----------------------------*

DROP TABLE IF EXISTS `submodule`;

CREATE TABLE `submodule` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(40) DEFAULT NULL,

`status` smallint(2) DEFAULT '1',

`parentModuleId` int(11) DEFAULT '100000',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=100009 DEFAULT CHARSET=utf8;

*-- ----------------------------*

*-- Records of submodule*

*-- ----------------------------*

INSERT INTO `submodule` VALUES ('100000', '留言管理', '1', '100000');

INSERT INTO `submodule` VALUES ('100001', '文章管理', '1', '100000');

INSERT INTO `submodule` VALUES ('100002', '审核管理', '1', '100000');

INSERT INTO `submodule` VALUES ('100003', '栏目管理', '1', '100000');

INSERT INTO `submodule` VALUES ('100004', '模块管理', '1', '100001');

INSERT INTO `submodule` VALUES ('100005', '角色分配', '1', '100001');

INSERT INTO `submodule` VALUES ('100006', '人员管理', '1', '100001');

INSERT INTO `submodule` VALUES ('100007', '角色管理', '1', '100001');

INSERT INTO `submodule` VALUES ('100008', '首页信息管理', '1', '100001');