## 新闻系统

## 系统需求分析和整体设计

1新闻系统的功能结构

一个新闻系统分前台和后台

前台需要显示新闻导航分类，新闻列表，新闻详情页，新闻搜索功能

后台需要管理管理员账号，管理新闻导航分类，新闻内容新增、编辑、删除功能

导航分类

前台页面

新闻列表

新闻详情

新闻搜索

后台页面

管理员管理

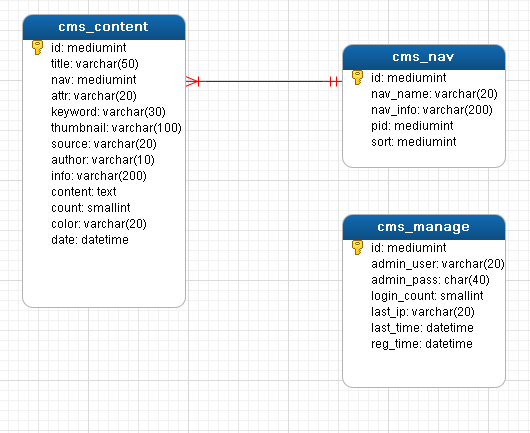
导航分类管理

新闻内容管理

管理员登录

**数据结构设计**

根据需求分析建立管理员账户信息表cms\_manage，新闻导航分类表cms\_nav，新闻内容表cms\_content。它们之间的关联图如下：

****

表结构：

管理员账户信息表cms\_manage

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 索引 | 注释 |
| Id | mediumint | 主索引 | 用户id |
| admin\_user | Varchar（20） |  | 账号 |
| admin\_pass | char(40) |  | 管理员密码 |
| login\_count | smallint(5) |  | 登录次数 |
| last\_ip | varchar(20) |  | 最后一次登录IP |
| last\_time | datetime |  | 最后一次登录时间 |
| reg\_time | datetime |  | 注册时间 |

新闻导航分类表cms\_nav

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 索引 | 注释 |
| Id | mediumint | 主索引 | 编号id |
| nav\_name | Varchar（20） |  | 导航名 |
| nav\_info | varchar (200) |  | 导航说明 |
| pid | mediumint(8) |  | 当该条记录是子分类时  该值为父id值 |
| sort | varchar(20) |  | 排序 |

新闻内容表cms\_content

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 索引 | 注释 |
| Id | mediumint | 主索引 | 编号id |
| title | Varchar（50） |  | 标题 |
| nav | mediumint(8) | 普通索引 | 导航分类id |
| attr | varchar(20) |  | 属性 |
| keyword | varchar(30) |  | 关键字 |
| thumbnail | varchar(100) |  | 缩略图 |
| source | varchar(20) |  | 文章来源 |
| author | varchar(10) |  | 作者 |
| info | varchar(200) |  | 内容摘要 |
| content | text |  | 内容 |
| count | smallint(6) |  | 浏览次数 |
| color | varchar(20) |  | 颜色 |
| date | datetime |  | 发布时间 |

**前台显示模块设计**

1前台显示模块主要功能是供用户浏览新闻。首页分三块：导航栏，搜索栏，四块分类新闻列表。如图：



2导航栏连接显示该分类下所有的新闻列表。显示内容：导航路径，新闻标题，新闻缩略图，新闻摘要 ，右侧子分类，分页。



3 新闻详情页



4 搜索列表 根据标题，关键字搜索。匹配字用红色标注



**后台显示模块设计**

1登录。账号，密码，验证码。登录时需验证

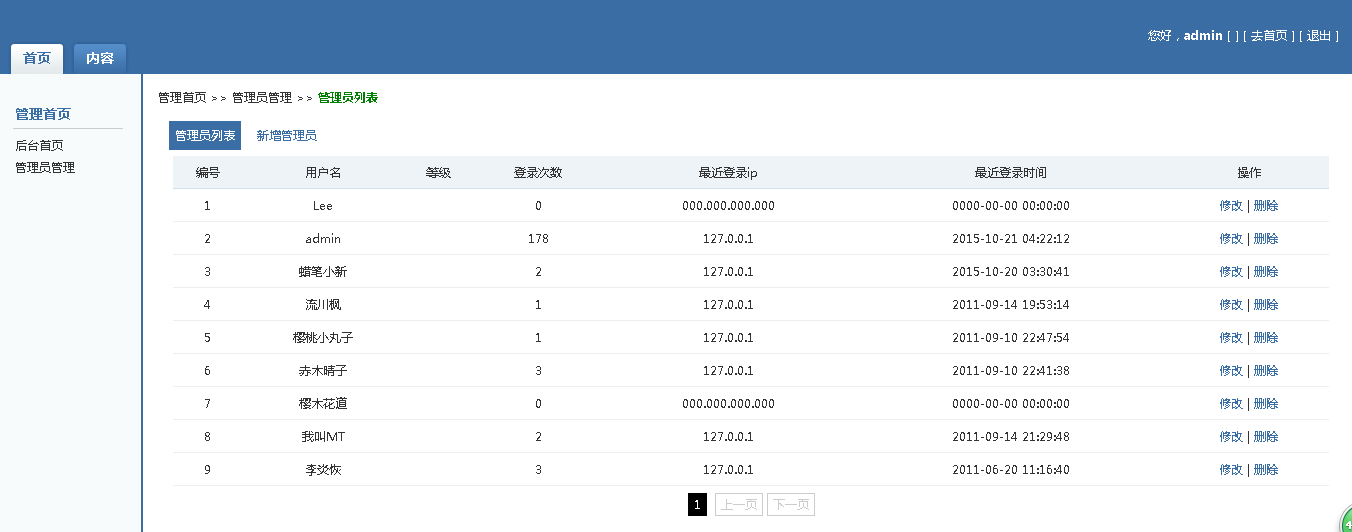


2 主页面分三部分（头部菜单，左侧菜单，主页面 ）

头部链接切换左侧菜单功能

左侧菜单链接切换主页面

默认为管理员管理列表页



3 新增导航



4新增修改新闻



## 程序开发设计：

## 三层架构设计

一． 什么是三层架构

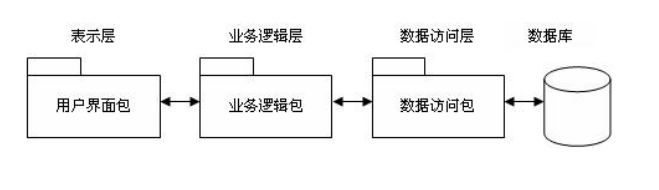
三层架构 (3-tier application) 通常意义上的三层架构就是将整个业务应用划分为：表现

层（ UI ） 、业务逻辑层（ BLL ） 、数据访问层（ DAL ） 。区分层次的目的即为了 “ 高内聚，低耦合 ” 的思想。

1 表现层 （ UI ） ： 通俗讲就是展现给用户的界面 ， 即用户在使用一个系统的时候他的所见所得。

2 业务逻辑层 （ BLL ） ： 针对具体问题的操作 ， 也可以说是对数据层的操作 ， 对数据业务逻辑处理。

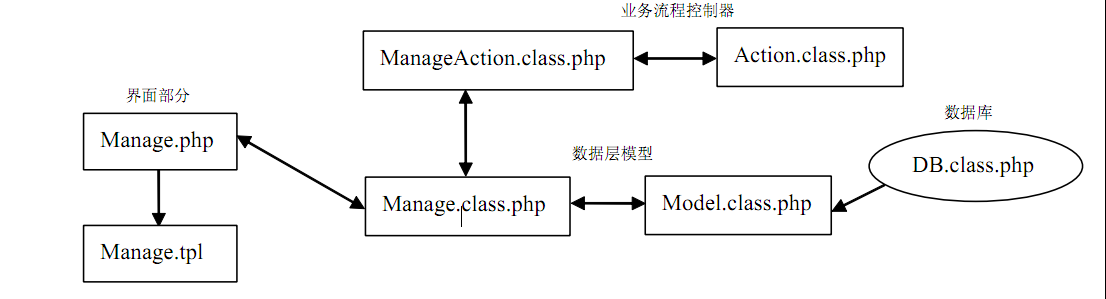
3 数据访问层（ DAL ） ：该层所做事务直接操作数据库，针对数据的增、删、改、查 。



二． 我们的项目是怎么分层的

CMS 项目采用了比较常规的三层架构的分层方式，我们用图例表示管理员 Manage 操作

细节：



三．目录及程序文件规范

Action/

action.class.php //控制器基类

LoginAction.class.php //后台登录控制器类

ManageAction.class.php //管理员管理控制器类

NavAction.class.php //后台导航分类管理控制器

ContentAction.class.php //后台管理新闻控制器类

DetailsAction.class.php //前台新闻详情控制器类

IndexAction.class.php //前台首页控制器类

ListAction.class.php //前台新闻列表控制器类

SearchAction.class.php //前台搜索列表控制器类

Model/

Model.class.php //模型基类

ContentModel.class.php //新闻表模型类

ManageModel.class.php //管理员表模型类

NavModel.class.php //导航分类表模型类

Config/ //配置目录

profile.inc.php //系统，数据库，模板 配置文件

ckeup.php //后台编辑器上传图片

includes/

Cookie.class.php //cookie类

DB.class.php //数据库连接类

FileUpload.class.php //上传文件类

Image.class.php //图像处理类

Page.class.php //分页类

Parser.class.php //模板解析类

Templates.class.php //模板类

Tool.class.php //常用工具类

Validate.class.php //验证数据类

ValidateCode.class.php //验证码类

Js/

Style/ //css样式

Ckeditor //新闻编辑器控件

Templates/ //模板目录

templates\_c/ //模板缓存

uploads/ //文件上传目录

admin/

admin.php //后台主程序

admin\_login.php //后台登陆页

code.php //验证码

content.php //新闻管理

manage.php //管理员管理

nav.php //导航分类管理

upload.php //上传缩略图

/init.inc.php //公共调用入口

/index.php //前台首页

/list.php //前台新闻列表

/search.php //搜索新闻列表

/details.php //新闻详情页

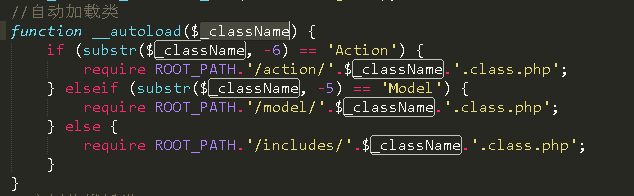
/common.inc.php //前台所需要的公共程序

1 配置信息 config/profile.inc.php

数据库设置，模板引擎目录配置，系统配置

在根目录下init.inc.php加载引入配置信息 require ROOT\_PATH.'/config/profile.inc.php';

使用魔术方法自动加载调用类



分离数据库连接

将数据库的连接方式分离到 includes 下 DB.class.php ，然后调用来使用。特别注意 ， 销毁数据集和关闭数据库连接的时候，采用传引用地址的方法。

$\_result 和 $\_db 不管是按值还是地址都可以清理和关闭。

而 $\_result 和 $\_db 对象必须在直接当前页清理，否则必须按照传地址的方式进行清理 。

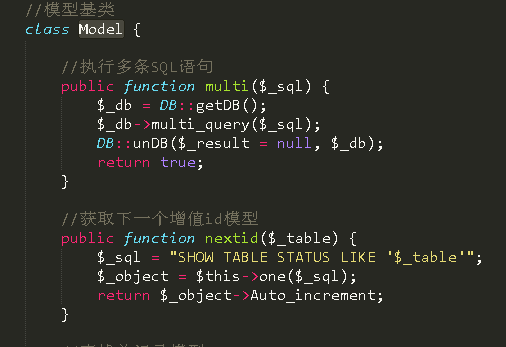
分离配置文件

数据库的连接配置信息，我们可以保存到 config/profile.inc.php 里。在系统初始化文件

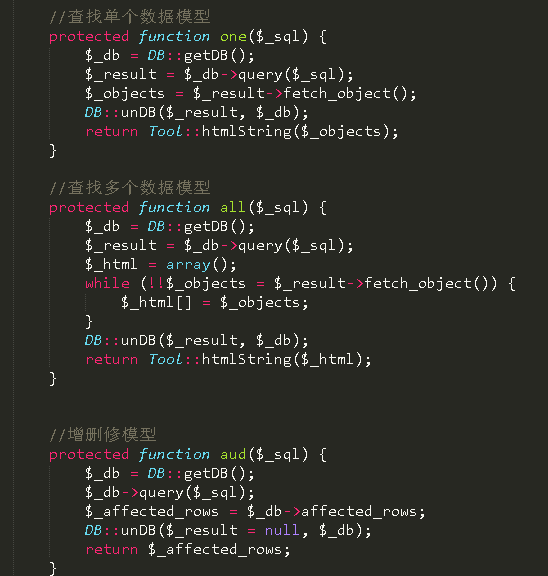
直接调用即可。

## 模型程序规范

1创建模型基类Model.class.php ，class名和文件命名相同



Model类创建 数据库增删改查等常用方法



2 新闻ContentModel.class.php 继承 model基类

私有属性对应新闻数据表字段

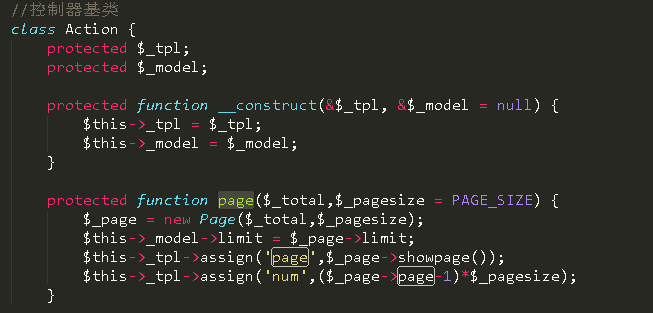
创建需要读取数据的方法



## 控制器程序规范

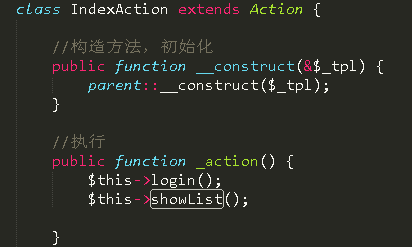
1创建控制器基类action.class.php ，class名和文件命名相同

2 构造方法需要使用模板对象和模型对象



3 首页控制器 IndexAction.class.php 继承action控制器基类

\_action方法作为执行业务逻辑



## 前台程序编写

1首页index.php

1.1首先加载公共调用入口require dirname(\_\_FILE\_\_).'/init.inc.php';

然后生成action 控制器对象 例如首页用的是IndexAction 控制器对象

$\_index = new IndexAction($\_tpl);并执行\_action

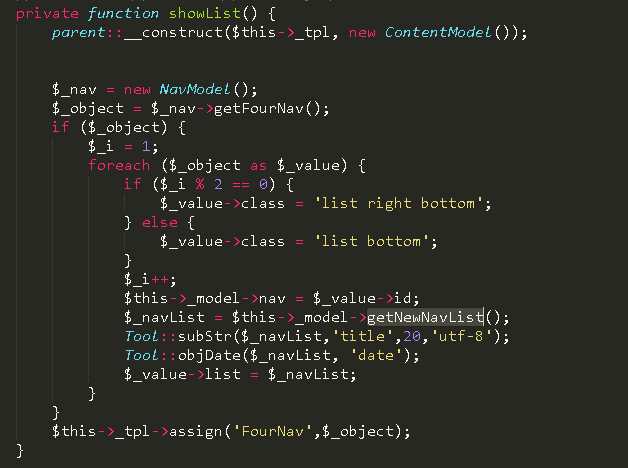
调用模板index.tpl 显示

1.2 控制器需要获取四块分类新闻列表数据

先获取前四个导航分类id值 然后根据id值获得最新的新闻。

需要使用新闻模型对象中的getNewNavList，导航分类对象getFourNav

在控制器中赋值给模板变量使用assign（变量名，值）方法实现

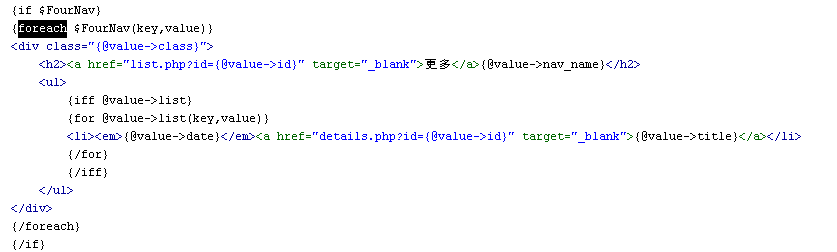


1.3 模板编写

头尾分离：由于各页面都需要调用同样的头部和尾部，需要把这里的代码分离开。

调用头部：{include file='header.tpl'}

显示数据用循环数组变量执行



2 搜索关键字新闻列表search.php

2.1用的是SearchAction 控制器对象

$\_search = new SearchAction($\_tpl));并执行\_action

调用模板search.tpl 显示

2.2 控制器需要获取用户输入的关键字来查询新闻

分页需要统计总数，分页记录数

需要使用新闻模型对象中的searchKeywordContentTotal, searchKeywordContent方法

根据标题，关键字搜索。匹配字用红色标注



## 后台程序编写

1 创建admin目录

我们要在目录中创建index.php文件为了防止直接输入admin目录报错

在index.php 增加跳转到admin.php 代码： Tool::alertLocation(null, 'admin.php')

在admin目录下要加载初始化程序文件我们可以用这代码：

require dirname(\_\_FILE\_\_).'./../init.inc.php';

2设计管理员

1. 超链接模式

|  |  |
| --- | --- |
| top菜单 | |
| 左侧菜单  sidebar | 主要页面Main |

2.1在框架页面 admin.tpl在左侧菜单设置名称 name="sidebar",在主要页面设置名称 name="main"

后台首页 top页面超链接 target=" sidebar "

<**a href="../templates/sidebarn.html" target="sidebar" id="nav2" onclick="***admin\_top\_nav*(2)**"**>内容</**a**>

2.2管理员页面 manage.php超链接 target="main"

<**a href="../admin/manage.php?action=show" target="main"**>管理员管理</**a**>

3． 创建 manage.php

3.1创建 ManageAction.class.php控制器继承action控制器

3.2创建 manage.php 调用控制器并执行方法

$\_manage = new ManageAction($\_tpl); //入口

$\_manage->\_action();

3.3创建manage.tpl ， 然后制作一个表格 ， 提供给个管理员数据列表的界面 。

th 背景色： #eef3f7;

th 下划线色： #d5dfe8;

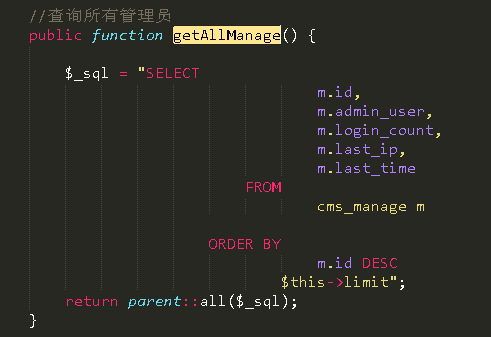
3.4． 创建 ManageModel.class.php在根目录创建 model 目录 ， 用于存放所有的实体类 ， 实体类里存放的是所有模块的数据

层操作：比如查询、新增、删除、修改等。

载入后，直接调用即可显示数据。

3.5． 显示所有数据

在实体类的 getAllManage () 方法里 ， 返回出二维的数组或对象数组 。 通过 parent::all($\_sql); 方法来获得数据值 。

****

添加修改操作

Sql语句 insert，update ，delete

提交数据时先使用 Validate检查验证数据是否正确。

如Validate::checkNull($\_POST['admin\_user']) 判断是否为空

验证错误使用Tool::alertBack方法警告提示如Tool::alertBack('警告：用户名不得为空！')

如新增管理员



4 新闻管理 content.php

4.1创建 ContentAction.class.php控制器继承action控制器

4.2 调用控制器并执行方法

$\_content = new ContentAction($\_tpl); //入口

$\_content->\_action();

4.3创建content.tpl ， 然后制作一个表格 ， 提供给个新闻列表的界面 。

添加修改页面通过ckeditor控件来进行文本编辑:调用js方法

<script type="text/javascript" src="../ckeditor/ckeditor.js"></script>

在使用编辑器的地方插入HTML控件

<textarea id="TextArea1" name="content" class="ckeditor"></textarea>

注意在控件中加上 class="ckeditor"。

配置编辑器/ckeditor/config.js

设置图片上传程序php路径

config.filebrowserImageUploadUrl = '../config/ckeup.php?type=img';

显示效果如下：

