CMS后端设计方案

目录

[一．CMS设计简介 1](#_Toc422150835)

[二．数据库设计 2](#_Toc422150836)

[社区数据库 2](#_Toc422150837)

[站内信管理 4](#_Toc422150838)

[三．前后端交互格式 5](#_Toc422150839)

# 一．CMS设计简介

CMS的功能模块图如下图所示，其中红框为一期要做的内容。

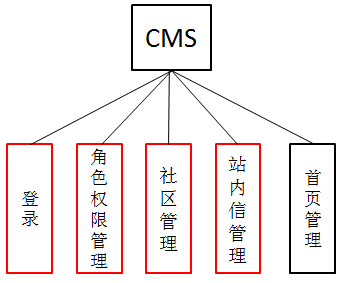


Figure 1 CMS 功能模块图

其中社区管理又分为一二期：

一期：帖子管理/评论管理/用户管理/举报管理/标签管理

二期：社区公告管理/Banner管理/推荐活动

CMS是独立于主站的另外一个站点，通过生产VPN访问，可以读写主站的数据服务，数据服务包括Mysql/Mongodb/静态存储服务器等，CMS的架构示意图如下图所示：

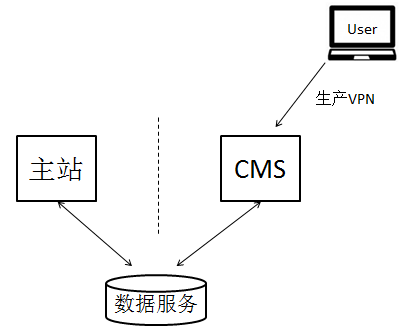


Figure 2 CMS架构示意图

CMS使用Django框架开发，社区管理和站内信管理都是其中的应用。

# 二．数据库设计

社区管理和站内信管理的数据服务只使用了Mysql，有些功能需要扩展主站中的数据库表，下面数据库设计中红色部分是新加入数据库的内容。

## 社区数据库

帖子表：

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| Hot\_count | Int(11) | 主题热度 |
| title | varchar(255) | 帖子标题 |
| Content | Text | 帖子内容的纯文本格式 |
| Status | Tinyint | 状态，有普通和取消两种状态 |
| Html\_content | text | 帖子内容的HTML格式 |
| Category | Int(11) | 帖子关联的分类信息 |
| Promotion\_order | Tinyint | 主页推荐帖子的显示顺序 |
| User\_id | Int(11) | 发帖用户ID |
| Created\_time | datetime | 创建时间 |
| Modified\_time | datetime | 修改时间 |

评论表：

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| Title | varchar(255) | 帖子标题 |
| Content | Text | 帖子内容的纯文本格式 |
| Status | Tinyint | 状态，有普通和取消两种状态 |
| Html\_content | text | 帖子内容的HTML格式 |
| Post\_id | Int(11) | 关联的帖子ID |
| Reference\_id | Int(11) | 评论关联的评论ID |
| Reply\_to\_id | Int(11) | 评论回复的关联用户ID |
| User\_id | Int(11) | 发表评论的用户ID |
| Created\_time | datetime | 创建时间 |
| Modified\_time | datetime | 修改时间 |

举报表：

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| Object\_type | varchar(255) | 关联的类型，目前为comment/post |
| Object\_id | Int(11) | 关联对象的id |
| Detail | Varchar(255) | 举报详情 |
| Status | Tinyint | 状态，有New和processed两种状态 |
| Reason | Tinyint | 举报原因，目前有1001/1002/1003/0几种 |
| Reporter | Int(11) | 发起举报的用户 |
| Created\_time | datetime | 创建时间 |
| Modified\_time | datetime | 修改时间 |

标签表：

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| Name | varchar(255) | 分类名称 |
| slug | varchar(255) | 分类简称，用于url |
| Created\_time | datetime | 创建时间 |
| Modified\_time | datetime | 修改时间 |

用户禁言表：

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| User\_id | varchar(255) | 用户ID |
| Report\_count | int(11) | 被举报次数 |
| Forbid | smallint(6) | 禁言类型 |
| Start\_time | datetime | 开始时间 |
| Day | Int(11) | 禁言天数 |

## 站内信管理

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 含义 |
| Id | int(11) | 主键，auto\_increment自增长 |
| User\_id | Int(11) | 接收站内信的用户ID |
| type | varchar(50) | 站内信类型 |
| status | Smallint(6) | 状态 |
| meta | longtext | 站内信的数据 |
| Created\_time | datetime | 创建时间 |
| Modified\_time | datetime | 修改时间 |

# 三．前后端交互格式

与前端数据传输使用json，举几个例子：

**帖子查询正常返回**：

{

"statusCode": 0,

"pages": {

"page": 1,

"totalPage": 3,

"recordsPerPage": 20,

"totalRecord": 55

},

"list": [

{"author": "妥妥828", "id": "128", "title": "【有奖征文】11111", "createDate": "2015/05/05 21:00:00", "isHide":true, "tags" : [

{"id": "1", "label": "推荐", "isChecked": true},

{"id": "2", "label": "加精置顶", "isChecked": false},

{"id": "3", "label": "笔记", "isChecked": false}

]},

{"author": "再见阳光", "id": "234", "title": "【有奖征文】33333", "createDate": "2015/05/05 21:00:00", "isHide":false, "tags" : [

{"id": "1", "label": "推荐", "isChecked": true},

{"id": "2", "label": "加精置顶", "isChecked": true},

{"id": "3", "label": "笔记", "isChecked": false}

]}

]

}

**查询结果为空**:

{

"statusCode": 0,

"pages":{

"page": 1,#前端传过来的值

"totalPage": 0,

"recordsPerPage": 10,#前端传过来的值

"totalRecord": 0

},

"list": []

}

**输入有误**：

{

"statusCode": -100,

"pages": {},

"list": []

}

**用户不存在**：

{

"statusCode": -1,

"pages": {},

"list": []

}

**帖子ID不存在**：

{

"statusCode": -2,

"pages": {},

"list": []

}