

# 博客项目实战

作者：少林之巅

# 目录

1. 项目目录规范
2. 功能模块划分
3. 数据库设计
4. 项目开发

## 项目目录划分

## 1. 目录划分介绍

# 项目目录划分

## 2. 各个目录详细介绍

A. config目录：用来存放项目的配置文件

B. dal目录：用来存放与数据库交互的相关代码。

C. logic目录：用来存放业务逻辑相关代码。

D. model目录：用来存放项目的数据结构相关代码。

E. static目录：用来存放项目的静态资源。

F. util目录：用来存放项目的一些工具功能以及杂项。

G. views目录：用来存放项目的一些工具功能以及杂项。

H. controller目录：用来存放控制器。

## 项目目录划分

### 3. MVC设计模式介绍

A. MVC全名： 模型（model）-视图（View）-控制器（controller）的缩写。

B. 是一种软件设计模式，用一种业务逻辑、数据、界面分离的代码组织方式。

C. Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。

D. View（视图）是应用程序中处理数据显示的部分。

E. Controller（控制器）是应用程序中处理用户交互的部分。

**优点：** MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

**耦合性低：** 视图层和业务逻辑层分离。相互独立，不受影响。

**重用性高：** 业务逻辑层可以完全复用。

**可维护性高：** 视图层和逻辑层分离，程序可维护性更好。

# 博客需求分析

## 4. 功能列表

- A. 文章列表，用户可以在首页看到已经发表的文章列表。
- B. 发表文章，用户可以投稿，发表文章，并展示在首页。
- C. 留言功能，用户可以在网站上进行留言，并展示在留言区。
- D. 点击排行，展示阅读次数最多的10篇文章。
- E. 评论排行，展示评论次数最多的10篇文章。
- F. 分类云，展示文章分类列表。

# 博客系统数据库设计

## 5. 文章表

```
CREATE TABLE `article` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '文章id',  
  `category_id` bigint(20) UNSIGNED NOT NULL COMMENT '分类id',  
  `content` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '文章内容',  
  `title` varchar(1024) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '文章标题',  
  `view_count` int(255) UNSIGNED NOT NULL COMMENT '阅读次数',  
  `comment_count` int(255) UNSIGNED NOT NULL COMMENT '评论次数',  
  `username` varchar(128) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '作者',  
  `status` int(10) UNSIGNED NOT NULL COMMENT '状态',  
  `summary` varchar(256) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '文章摘要',  
  `create_time` timestamp(0) NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '发布时间',  
  `update_time` timestamp(0) NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP(0) COMMENT '更新时间',  
  PRIMARY KEY (`id`) USING BTREE,  
  INDEX `idx_view_count` (`view_count`) USING BTREE COMMENT '阅读次数索引',  
  INDEX `idx_comment_count` (`comment_count`) USING BTREE COMMENT '评论数索引',  
  INDEX `idx_category_id` (`category_id`) USING BTREE COMMENT '分类id索引'  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;
```

# 博客系统数据库设计

## 6. 文章分类表

```
-- -----  
-- Table structure for cateogry  
-- -----  
DROP TABLE IF EXISTS `cateogry`;  
CREATE TABLE `cateogry` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `category_name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '分类',  
  `category_no` int(10) UNSIGNED NOT NULL COMMENT '分类排序',  
  `create_time` timestamp(0) NULL DEFAULT CURRENT_TIMESTAMP,  
  `update_time` timestamp(0) NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP(0),  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;  
-- -----
```



# 博客系统数据库设计

## 7. 评论表

```
-- -----  
-- Table structure for comment  
-- -----  
DROP TABLE IF EXISTS `comment`;  
CREATE TABLE `comment` (  
  `id` bigint(20) UNSIGNED NOT NULL COMMENT '评论id',  
  `content` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '评论内容',  
  `username` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '评论作者',  
  `create_time` timestamp(0) NOT NULL ON UPDATE CURRENT_TIMESTAMP(0) COMMENT '评论发布时间',  
  `status` int(255) UNSIGNED NOT NULL COMMENT '评论状态: 0, 删除; 1, 正常',  
  `article_id` bigint(20) UNSIGNED NULL DEFAULT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;
```

# 博客系统数据库设计

## 8. 留言表

```
-- -----  
-- Table structure for leave  
-- -----  
DROP TABLE IF EXISTS `leave`;  
CREATE TABLE `leave` (  
  `id` bigint(20) NOT NULL,  
  `username` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  `email` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  `content` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  `create_time` timestamp(0) NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `update_time` timestamp(0) NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP(0),  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_general_ci ROW_FORMAT = Dynamic;
```

# 数据实体设计

## 9. 文章实体

```
type ArticleInfo struct {  
    Id          int64      `db:"id"`  
    CategoryId  int64      `db:"category_id"`  
    Summary     string     `db:"summary"`  
    Title       string     `db:"title"`  
    ViewCount   uint32     `db:"view_count"`  
    CreateTime  time.Time  `db:"create_time"`  
    CommentCount uint32     `db:"comment_count"`  
    Username    string     `db:"username"`  
}  
  
type ArticleDetail struct {  
    ArticleInfo  
    Content string `db:"content"`  
    Category  
}  
  
type ArticleRecord struct {  
    ArticleInfo  
    Category  
}
```

## 数据实体设计

### 10. 文章分类实体

```
type Category struct {  
    CategoryId  int64 `db:"id"`  
    CategoryName string `db:"category_name"`  
    CategoryNo  int    `db:"category_no"`  
}
```

# 数据实体设计

## 11. 评论实体

```
type Comment struct {  
    Id          int64      `db:"id"`  
    Content     string     `db:"content"`  
    Username    string     `db:"username"`  
    CreateTime  time.Time  `db:"create_time"`  
    Status      int        `db:"status"`  
    ArticleId   int64      `db:"article_id"`  
}
```

## 文章模块开发

### 11. 数据层开发

- A. 文章插入接口。
- B. 获取文章列表接口。
- C. 获取文章分类接口。
- D. 获取文章详情接口。

## 文章模块开发

### 12. 业务逻辑层开发

- A. 文章插入接口。
- B. 获取文章列表接口。
- C. 获取文章分类接口。
- D. 获取文章详情接口。

## 文章模块开发

### 13. Controller层开发

- A. 文章列表渲染。
- B. 文章详情页渲染。
- C. 文章发布页渲染。