|  |
| --- |
|  |
| 高性能内容发布网站系统探究 |
| 中大毕业设计大纲 |
|  |
| **ming123jew** |
| **2018-2-5** |

|  |
| --- |
|  |

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| 2018-02-05 | 1.00 | 提纲 | ming123jew |
| 2018-03-31 | 1.01 | 论文初稿（少部分未完成） | ming123jew |
| 2018-04-03 | 1.02 |  |  |

目录

[一、 引言 5](#_Toc510545355)

[(一) 研究背景 5](#_Toc510545356)

[(二) 研究目的 6](#_Toc510545357)

[(三) 可行性分析 6](#_Toc510545358)

[1. 经济方面 6](#_Toc510545359)

[2. 技术方面 6](#_Toc510545360)

[3. 管理方面 6](#_Toc510545361)

[二、 技术框架、运行环境与开发工具 6](#_Toc510545362)

[(一) 技术框架 6](#_Toc510545363)

[1. HTML前端框架之Bootstrap 6](#_Toc510545364)

[2. PHP服务框架之swoole 7](#_Toc510545365)

[3. SwooleDistributed框架 7](#_Toc510545366)

[(二) 运行环境 8](#_Toc510545367)

[1. centos7.4服务器 8](#_Toc510545368)

[2. 超文本预处理器之PHP7 9](#_Toc510545369)

[3. git版本管理工具 10](#_Toc510545370)

[4. 间约客户端hiredis 10](#_Toc510545371)

[5. mysql服务器 10](#_Toc510545372)

[6. 测试运行环境 10](#_Toc510545373)

[二、 系统设计 11](#_Toc510545374)

[(一) MVC架构 11](#_Toc510545375)

[(二) 系统安全 11](#_Toc510545376)

[1. centos服务器安全 11](#_Toc510545377)

[2. 系统服务进程安全 11](#_Toc510545378)

[(三) 模块安全设计 11](#_Toc510545379)

[1. 防止内存溢出 11](#_Toc510545380)

[2. 参数过滤 12](#_Toc510545381)

[3. 数据库安全 12](#_Toc510545382)

[三、 功能模块 12](#_Toc510545383)

[(一) 网站后台模块 12](#_Toc510545384)

[1. 登录 12](#_Toc510545385)

[2. 角色权限管理 13](#_Toc510545386)

[3. 用户管理 15](#_Toc510545387)

[4. 栏目管理 16](#_Toc510545388)

[5. 内容管理 17](#_Toc510545389)

[6. 后台主页 19](#_Toc510545390)

[(二) 网站前台模块 19](#_Toc510545391)

[1. 首页 19](#_Toc510545392)

[2. 栏目页 19](#_Toc510545393)

[3. 内容页 19](#_Toc510545394)

[4. 评论 19](#_Toc510545395)

[5. 点赞 19](#_Toc510545396)

[6. 个人简介（单页） 19](#_Toc510545397)

[(三) 设计 19](#_Toc510545398)

[(四) 网站访问统计 19](#_Toc510545399)

[三、 总结 19](#_Toc510545400)

[[1]百度百科.词条PHP（超文本预处理器）[引用日期2018-02-05] 19](#_Toc510545401)

[[2]SwooleDistributed官网.SwooleDistributed定义[引用日期2018-02-05] 19](#_Toc510545402)

[[3]分布式版本控制系统 Git ．开源社区网[引用日期2012-09-25] 19](#_Toc510545403)

高性能内容发布网站系统探究

[摘要] PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。[1]本项目就PHP编程语言之扩展框架（SwooleDistributed[2]，下文简称SD）进行高性能内容发布网站系统（偏重个人博客）探究，研究SD框架的性能表现，以及开发流程以及使用各个细节，最终完成一套属于本人甚至企业使用的高性能内容管理系统。

[关键词] PHP、SD、高性能、内容发布 、并发、功能模块

1. 引言
   1. 研究背景

21世纪以来，由于计算机技术和网络技术迅猛发展，世界已经进入了信息社会。随着个人上网用户和企事业单位用户上网的增加，并且近年来电子商务的流行，使得人们越来越依赖网络。越来越多的企业和个体成员投入到互联网行列中，如企业官方网站、个人博客、社交平台、电子商务等平台。多种多样性质的网站推动着开发网站技术的发展，由初期多数基于asp编程技术建设的网站，逐步转向PHP、.NET、JSP等语言平台编程方向，随着互联网的不断发展，对大数据的需求，特别部分社交平台对访问并发技术的需求量也是越来越大，逐渐服务架构+编程语言的并发能力越来越得到重视。本设计就PHP编程语言之扩展框架（SwooleDistributed）进行高性能内容发布网站系统（偏重博客）探究。

* 1. 研究目的

现在各行各业都需要利用网站作为宣传的媒体，因此网站的整体需求是很大的，网站内容管理系统的前景也是很广阔，特别高性能内容管理系统，无论在并发访问量或降低企业硬件成本，或者技术需求上更是受到媒体以及互联网的青睐。高性能内容管理系统成为我国各行各业、各领域的基础设施建设重点。在这种大趋势下，无论是在理论上还是在应用上，对于一个标准的完整的高性能内容管理系统的研究与应用都是具有重要意义的。基于此，本项目研究的目的就PHP编程语言之扩展框架（SwooleDistributed）完成对网站内容管理系统中的角色管理、栏目管理、内容发布等核心功能，实现一个属于自己/企业的高性能内容管理系统。

* 1. 可行性分析

可行性分析(Feasibility Analysis)也称为可行性研究，是在系统调查的基础上，针对新系统的开发是否具备必要性和可能性，对新系统的开发从技术、经济、社会的方面进行分析和研究，以避免投资失误，保证新系统的开发成功。可行性研究的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。

* + 1. 经济方面

本系统对系统开发者来说并不需要太高的成本支出，只是对系统的管理者付出管理报酬即可，而且开发周期不需要太长，节省了人力、物力、财力资源，所以本系统在经济上是可行的。

* + 1. 技术方面

在软件方面，本系统采用编程语言PHP(PHP Hypertext Preprocessor)进行开发，前端网页框架使用的是Bootstrap框架，而数据库系统采用的是MYSQL，相关技术都是目前较为流行且开源的技术。通过多方面分析，在软、硬件方面现有工具与环境完全可以实现系统的开发，因此具有技术上的可行性。目前越来越多的应用程序都已经是转向基于Web的开发，并且Internet已经广泛使用，因此系统的设计具有时机可行性。

* + 1. 管理方面

系统的开发主要就是为了方便个人信息的管理，补充现在传统管理方式的不足，因此具备了管理上的可行性。

综上所述以及结合当前背景情况，本系统开发目标已明确，在技术和经济等方面具备可行性，并且投入少、见效快，因此系统的开发是完全可行的。

1. 技术框架、运行环境与开发工具
   1. 技术框架
      1. HTML前端框架之Bootstrap
         1. Bootstrap介绍

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JavaScript 的，它简洁灵活，使得 Web 开发更加快捷。 [1] 它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。 [2] 国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

* + - 1. Bootstrap Admin Templates

目前为止最漂亮的Bootstrap后台模板之一，支持屏幕自适应，UI丰富多彩，控制完整，包括各种表单控件、图表控制、表格控件，支持从右向左的阅读习惯，支持多级菜单，提供表单编辑器，动画效果酷炫非凡。本系统程序后台搭建用此框架进行后台制作可减免后台页面设计工作，大大提高后台页面制作效率。

* + 1. PHP服务框架之swoole
       1. swoole框架介绍

Swoole是一种PHP高级Web开发框架，框架不是为了提升网站的性能，是为了提升网站的开发效率。最少的性能损耗，换取最大的开发效率。利用Swoole框架，开发一个复杂的Web功能，可以在很短的时间内完成了。

* + - 1. swoole框架特点
  + 类似ORM的数据查询，提供SQL封装器，让MySQL的SQL与PHP的Array，会话，Cache无缝结合。
  + MVC分层结构，有效的程序结构分层，提高程序的可维护性和扩展性，实现低耦合，基于接口开发。
  + 集成大量，实用的功能，比如方便的数据库操作，模板操作，缓存操作，系统配置，表单处理，分页，数据调用，字典操作，上传处理，内容编辑，调试等。
  + 模板-数据反射系统，可以直接在模板中调用数据，提供很多标签，可是无需修改程序，只修改模板，即可实现网站各类更新维护工作。
  + Swoole包含了大量类，提供众多的功能扩展，基本上Web开发能够用到的功能类，大部分都可以在Swoole框架中找到。
  + Swoole拥有插件系统，Fckeditor、Adodb、pscws中文分词、中文全文索引系统、最新的Key-Value数据库思想，TokyoTyrant，可以无限扩展框架的功能。
    1. SwooleDistributed框架
       1. SwooleDistributed框架介绍

SD框架全称SwooleDistributed，从名称上看一个是Swoole一个是Distributed，他是基于Swoole扩展的可以分布式部署的应用服务器框架。 借助于PHP的高效开发环境，Swoole的高性能异步网络通信引擎，以及其他的高可用的扩展和工具，SD框架提供给广大开发者一个稳定的高效的而且功能强大的应用服务器框架。

* + - 1. SwooleDistributed框架特点
  + 混合协议 SD框架支持长连接协议TCP，WebSocket，短连接协议HTTP，以及UDP。 通过配置开放不同的端口开发者可以轻松管理不同的协议，并且可以共用一套业务代码，当然你可以通过智能路由进行代码的隔离。 长连接可以配置不同的数据传输协议，比如二进制协议文本协议等等，通过框架提供的封装器解包器接口可以自定义各种各种的协议封装，并且各种协议之间可以自动转换，比如你通过广播发送一个信息，该信息流向不同客户端，客户端间采用不同协议，那么框架会根据不同的端口自动转换不同的协议封装。 你也可以通过Http给所有长连接客户端发送推送消息，类似这种混合协议协作的业务在SD框架上会异常简单。
  + MVC以及智能路由 框架的设计是MVC架构，其中每一个层级都可以继续划分子层级，开发者可以将Controller继续分层通过不同文件夹进行管理，也可以将Model进行细分，划分为业务层和数据层，这都看开发者自身的系统设计。智能路由将处理解包器解包后的数据，负责将这些数据传递到Controller层。
  + 中间件 SD框架还提供了中间件，中间件可以对流入的数据进行处理，比如清理异常数据，修改数据，流量统计，搜集日志等功能。中间件可以设置多个，他们和端口进行绑定。
  + 对象池 SD框架内大多数的对象都使用了对象池技术，对象池技术有利于系统内存的稳定，减少GC的次数，提高系统的运行效率，事实证明对象池对系统稳定做出了极大的贡献，开发者也可以使用这一套对象池技术，增加对对象的复用，减少GC和NEW的频率，对系统毛刺现象和内存泄露方面都有很大的稳定性提升。
  + 异步客户端以及连接池 Mysql，Redis，Http客户端，Tcp客户端，等等其他更为复杂的客户端，在SD框架中均为异步的模式，异步解决了系统整体的并发能力，但异步客户端需要提供连接池维持，SD框架提供了连接池，开发者不需要自己管理连接池，只需要使用即可。
  + 协程、异步事件回调解决的是并发性能，但造成的是业务代码的混乱。SD框架提供了协程解决了这一问题，通过yield关键字提供对异步的同步写法，消除了业务书写上的大量回调嵌套，你可以通过yield+同步的写法实现异步的性能。 协程提供了一整套完整的体系，包括超时，异常，休眠，多路选择，以及创建用户协程等等功能。
  + 定时任务 顾名思义定时执行的任务。
  + 任务投递 支持将耗时任务投递到Task进程。
  + 自动Reload 可以开启框架的自动Reload功能，这样代码修改会被立即响应。
  1. 运行环境
     1. centos7.4服务器

CentOS（Community Enterprise Operating System，中文意思是：社区企业操作系统）是Linux发行版之一，它是来自于Red Hat Enterprise Linux依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，因此有些要求高度稳定性的服务器以CentOS替代商业版的Red Hat Enterprise Linux使用。两者的不同，在于CentOS并不包含封闭源代码软件。

CentOS 是一个基于Red Hat Linux 提供的可自由使用源代码的企业级Linux发行版本。每个版本的 CentOS都会获得十年的支持（通过安全更新方式）。新版本的 CentOS 大约每两年发行一次，而每个版本的 CentOS 会定期（大概每六个月）更新一次，以便支持新的硬件。这样，建立一个安全、低维护、稳定、高预测性、高重复性的 Linux 环境。 [1] CentOS是Community Enterprise Operating System的缩写。

CentOS 是RHEL（Red Hat Enterprise Linux）源代码再编译的产物，而且在RHEL的基础上修正了不少已知的 Bug ，相对于其他 Linux 发行版，其稳定性值得信赖。CentOS在2014初，宣布加入Red Hat。

* + 1. 超文本预处理器之PHP7
       1. PHP7介绍

PHP（外文名:PHP Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。

PHP语法吸收了C语言、Java和Perl的特点，易于学习，使用广泛，主要适用于Web开发领域。PHP 独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快的效果。

* + - 1. PHP7特点

PHP7要打破一切。 PHP开发人员应该接受打破版本之间向下兼容的定律。只要不允许大量的向后兼容，PHP7将是一个高度尊重的语言。

* + 创建一个具体的核心语言 删除所有库方法，并保持在对象集中的核心方法。 您应该能够编写无需任何外部库或扩展PHP7和对基本输入/输出，字符串处理和数学一个很好的完整的语言。库以外的任何应该通过批准扩展。
  + 一切都当作一个对象 以从Ruby，Smalltalk和(主要)的Java对象，并把它一切当作对象。 整数是对象，字符串是对象，他们每个人都可以操作的方法， 我不相信PHP需要的Ruby和Smalltalk在对象之间传递彼此讯息的观念，而调用对象的方法才是最好的。
  + 一致的命名方法和类 由于PHP的最大的抱怨之一是不断要检查，(needle,haystack) 或(haystack, needle)，或some\_function()，或function\_some()，或someFunction()，一个一致的格式需要制定。
  + 让事情严格尝试传递到一个方法浮动字符串? 这是一个警告。
  + 一切是Unicode 在PHP6中的所有字符串都是Unicode，这很好，我主张PHP7也应该保持。
  + 中央启动点 创建一个主类或初始化，所有代码执行源于此。
  + 清理C代码我不是一个C的专家，但如果你比较了解Ruby的C代码到PHP的C代码，可以很容易地了解了PHP与Ruby的内部。 我非常熟悉PHP，所以我自己的写扩展更容易。
  + 摆脱eval() eval()是邪恶的。 如果你正在使用它，那么这是一个错的主意：这将打破PHPUnit，抛弃它从现在开始。
  + 支持操作符重载 因为一切都是对象，开发者只需掌握操作对象的方法即可。
  + 允许的方法签名
    - 1. 内存工具redis-3.0.8

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持。从2013年5月开始，Redis的开发由Pivotal赞助。

* + 1. git版本管理工具
       1. git介绍

git(读音为/gɪt/。)是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

git 是用于 Linux内核开发的版本控制工具。与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持（wingeddevil注：这得分是用什么样的服务端，使用http协议或者git协议等不太一样。并且在push和pull的时候和服务器端还是有交互的。），使源代码的发布和交流极其方便。 Git 的速度很快，这对于诸如 Linux kernel 这样的大项目来说自然很重要。 git 最为出色的是它的合并跟踪（merge tracing）能力。

* + - 1. git版本工具安装

本系统程序使用git版本工具进行管理，其系统相关的框架以及部分相关组件均在github平台上面。因此，git作为本系统开发相关的必备工具。登录centos，在控制台中输入命令，yum install git进行自动安装。

* + 1. 间约客户端hiredis
       1. hiredis介绍

Hiredis是Redis数据库的简约C客户端库。它是简约的，因为它只是增加了对协议的 最小支持，但是同时它使用了一个高级别的printf-like API，所以对于习惯了printf风格的C编程用户来说，其非常容易使用，而且API中没有明确的绑定每个Redis命令。本系统sd框架启用redis模式运行系统，需要以来hiredis工具支持。

* + - 1. hiredis安装

由于安装过程中，hiredis安装经常容易出现出错，因此在本论文作安装记录。

git clone https://github.com/redis/hiredis.git

make -j

sudo make install

sudo ldconfig

/etc/ld.so.conf文件，在新的一行中加入库文件所在目录；

**备注：**如php\_error提示:

[15-Nov-2017 05:48:46 UTC] PHP Warning: PHP Startup: Unable to load dynamic library '/home/work/php-7.1.0/lib/php/extensions/no-debug-non-zts-20160303/swoole.so' - libhiredis.so.0.13: cannot open shared object file: No such file or directory in Unknown on line 0

则加入libhiredis.so.0.13所在路径.

* + 1. mysql服务器
    2. 测试运行环境

测试环境是否安装成功：

php –m 看其模块是否包括redis bcmath swoole，如不出现，则需要重新部署环境。

1. 系统设计
   1. MVC架构

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

其中这里需要特别说明，系统中一部分较为复杂的数据表（如内容表sd\_conteng），其Model层又细分为数据逻辑层与数据访问层，以简洁控制器代码编写，同时更好的实现“高内聚低耦合”的思想。

业务逻辑层：主要是针对具体的问题的操作，也可以理解成对数据层的操作，对数据业务逻辑处理，如果说数据层是积木，那逻辑层就是对这些积木的搭建。

数据访问层：主要是对非原始数据（数据库或者文本文件等存放数据的形式）的操作层，而不是指原始数据，也就是说，是对数据库的操作，而不是数据，具体为业务逻辑层或表示层提供数据服务。

* 1. 系统安全
     1. centos服务器安全

服务器安全作为系统运行的基本条件，只有对服务器进行安全加固以及不断的优化，运行在该系统的程序才能更加稳定、快速。本毕业设计只要探讨“高性能内容发布网站系统”这个主题，因此在此不进行系统安全方面的详细介绍。

* + 1. 系统服务进程安全

除了服务器的安全问题，其程序相关的服务安全同意非常重要。一个系统的正常运行往往需要多种不同的服务支持，如本系统程序运行，需要依赖redis服务、mysql服务、自身服务等。本系统为了方便管理上述相关服务进程，引入Supervisor工具进行管理。

Supervisor是用Python开发的一套通用的进程管理程序，能将一个普通的命令行进程变为后台daemon，并监控进程状态，异常退出时能自动重启。它是通过fork/exec的方式把这些被管理的进程当作supervisor的子进程来启动，这样只要在supervisor的配置文件中，把要管理的进程的可执行文件的路径写进去即可。也实现当子进程挂掉的时候，父进程可以准确获取子进程挂掉的信息的，可以选择是否自己启动和报警。supervisor还提供了一个功能，可以为supervisord或者每个子进程，设置一个非root的user，这个user就可以管理它对应的进程。

* 1. 模块安全设计
     1. 防止内存溢出

系统中可能会遇到大数组，其操作会导致系统发生内存溢出，因此在处理大数组时，设置PHP使用内存键值[ini\_set('memory\_limit','256M');]，同时设置超时限制[set\_time\_limit(600);]，另外，在使用完大数组后，使用unset对其进行销毁。

避免读取大文件，如读取大文件操作，则使用PHP的fopen，fgets函数进行指针操作，或使用SplFileObject文件操作类。

递归处理，在递归处理方面，需要注意无限递归/循环导致内存溢出，递归函数逻辑需要非常清晰。

* + 1. 参数过滤

SD框架在Controller父类中对其进行实例化，其功能主要是对原生swoole\_http\_request对象进行包装，让开发者可以轻松的获取get、post、header、cookie、file等值。“高性能内容发布网站系统”基于SD框架进行架构开发，因此，在参数过滤方面续承了统一处理，轻松过滤用户提交的相关参数，提高系统安全。

* + 1. 数据库安全

程序中的所有错误信息必须关闭或者屏蔽，防止系统信息涉漏。

服务器端的magic\_quotes，虽然他们默认都是打开的（magic\_quotes\_gpc），但仍然可能存在被恶意注入，因此主动关闭magic\_quotes\_gpc，全部自己手工对所有变量添加magic\_quotes，就是添加addslashes进行过滤。

1. 功能模块
   1. 网站后台模块
      1. 登录
         1. 流程图

帐号+密码

验证

后台

y

n

提示

用户在未登录状态下进行后台相关页面或接口访问，后台程序会自动检测管理员是否已经登录，如未登录则跳转到登录界面，提示用户输入账号和密码进行登录。后台程序开放了登录控制器的检测，因此登录界面可无条件进行访问。

用户输入账号和密码后，前端使用JavaScript进行简单的验证操作，如系统对账号是否为空、密码是否为空，长度等做了简单的验证，JavaScript验证通过后，提交到服务器进行验证登录。后台的登录验证功能主要分为用户登录时的身份认证与操作后台功能权限认证。用户提交登录后，程序首先会进对数据进行过滤操作，然后进行身份认证操作，所输入账号与密码能够与数据库查询到的数据相匹配，则认证成功，否则提示相关登录失败信息。如登录成功，程序会把认证成功的用户信息保存到服务器session，然后进行权限认证操作。

权限认证详细介绍将在角色权限进行介绍。

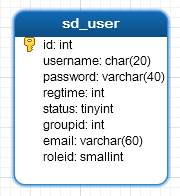
* + - 1. 数据字典

sd\_user表:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| id | int | 11 |  | y |  |  |  |
| username | char | 20 |  |  |  |  |  |
| password | varchar | 40 |  |  |  |  |  |
| regtime | int | 11 |  |  |  |  |  |
| status | tinyint | 2 |  |  |  |  |  |
| groupid | int | 8 |  |  |  |  |  |
| email | varchar | 60 |  |  |  |  |  |
| roleid | smallint | 6 |  |  |  |  |  |

* + - 1. 数据表模型

sd\_user表:



* + 1. 角色权限管理
       1. 流程图

后台URL

模块名+控制器名+方法名

用户信息(role\_id)

所在角色权限

验证

执行

y

提示

n

权限认证（RBAC），基于角色的权限访问控制（Role-Based Access Control）作为传统访问控制（自主访问，强制访问）的有前景的代替受到广泛的关注。在RBAC中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。在一个组织中，角色是为了完成各种工作而创造，用户则依据它的责任和资格来被指派相应的角色，用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限，而权限也可根据需要而从某角色中回收。角色与角色的关系可以建立起来以囊括更广泛的客观情况。

RBAC支持三个著名的安全原则：最小权限原则，责任分离原则和数据抽象原则。

* + - 最小权限原则之所以被RBAC所支持，是因为RBAC可以将其角色配置成其完成任务所需要的最小的权限集。
    - 责任分离原则可以通过调用相互独立互斥的角色来共同完成敏感的任务而体现，比如要求一个计帐员和财务管理员共参与同一过帐。
    - 数据抽象可以通过权限的抽象来体现，如财务操作用借款、存款等抽象权限，而不用操作系统提供的典型的读、写、执行权限。然而这些原则必须通过RBAC各部件的详细配置才能得以体现。
      1. 数据字典

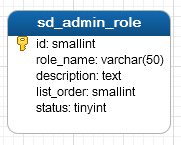
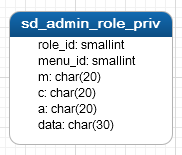
sd\_admin\_role表:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| id | smallint | 6 |  | y |  |  |  |
| role\_name | varchar | 50 |  |  |  |  |  |
| descript | text |  |  |  |  |  |  |
| list\_order | smallint | 5 |  |  |  |  |  |
| status | tinyint | 2 |  |  |  |  | 排序 |

sd\_admin\_role\_priv表:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| role\_id | smallint | 6 |  |  |  | 关联sd\_admin\_role表 |  |
| menu\_id | smallint | 6 |  |  |  | 关联sd\_admin\_menu表 |  |
| m | char | 20 |  |  |  |  |  |
| c | char | 20 |  |  |  |  |  |
| a | char | 20 |  |  |  |  |  |
| data | char | 30 |  |  |  |  |  |

* + - 1. 数据表模型

* + 1. 用户管理
       1. 功能图

用户列表

角色列表

添加用户

权限设置

删除权限

添加角色

添加用户

用户

用户管理

角色权限

* + - 1. 数据字典

请查阅登陆模块中的数据字典。

* + - 1. 数据表模型

请查阅登陆模块中的数据表模型。

* + 1. 栏目管理
       1. 功能图

栏目列表

添加栏目

编辑栏目

删除栏目

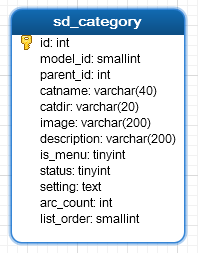
栏目管理

* + - 1. 数据字典

sd\_category表:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| id | int | 11 |  | y |  |  |  |
| model\_id | smallint | 5 |  |  |  |  |  |
| parent\_id | int | 11 |  |  |  |  |  |
| catname | varchar | 40 |  |  |  |  |  |
| catdir | varchar | 20 |  |  |  |  |  |
| image | varchar | 200 |  |  |  |  |  |
| description | varchar | 200 |  |  |  |  |  |
| is\_menu | tinyint | 2 |  |  |  |  |  |
| status | tinyint | 2 |  |  |  |  |  |
| setting | text |  |  |  |  |  |  |
| arc\_count | int | 11 |  |  |  | 文章数 |  |
| list\_order | smallint | 5 |  |  |  | 排序 |  |

* + - 1. 数据表模型



* + 1. 内容管理
       1. 功能图

内容列表

添加内容

编辑内容

删除内容

内容管理

* + - 1. 数据字典

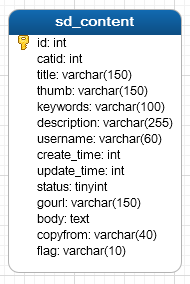
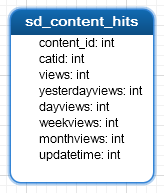
sd\_content表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| id | int | 11 |  | y |  |  |  |
| catid | int | 11 |  |  |  | 栏目ID |  |
| title | varchar | 150 |  |  |  |  |  |
| thumb | varchar | 150 |  |  |  |  |  |
| keywords | varchar | 100 |  |  |  |  |  |
| description | varchar | 255 |  |  |  |  |  |
| username | varchar | 60 |  |  |  | 发布者 |  |
| create\_time | int | 11 |  |  |  |  |  |
| update\_time | int | 11 |  |  |  |  |  |
| status | tinyint | 2 |  |  |  |  |  |
| gourl | varchar | 150 |  |  |  | 跳转链接 |  |
| body | text |  |  |  |  | 内容 |  |
| copyfrom | varchar | 40 |  |  |  | 来源 |  |
| flag | varchar | 10 |  |  |  | 推荐标识 |  |

sd\_content\_hits表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 空值 | 主键索引 | 默认值 | 说明 | 备注 |
| content\_id | int | 11 |  |  |  |  | 关联sd\_content表 |
| catid | int | 11 |  |  |  |  | 关联sd\_category表 |
| views | int | 10 |  |  |  |  |  |
| yesterdayviews | int | 10 |  |  |  |  |  |
| dayviews | int | 10 |  |  |  |  |  |
| weekviews | int | 10 |  |  |  |  |  |
| monthviews | int | 10 |  |  |  |  |  |
| updatetime | int | 11 |  |  |  |  |  |

* + - 1. 数据表模型

* + 1. 后台主页
  1. 网站前台模块
     1. 首页
     2. 栏目页
     3. 内容页
     4. 评论
     5. 点赞
     6. 个人简介（单页）
  2. 设计
  3. 网站访问统计

1. 总结

参考资料：

[1][百度百科](https://baike.baidu.com/).[词条PHP（超文本预处理器）](https://baike.baidu.com/item/PHP/9337?fr=aladdin)[引用日期2018-02-05]

[2][SwooleDistributed官网](http://sd.youwoxing.net/).SwooleDistributed定义[引用日期2018-02-05]

[3]分布式版本控制系统 Git ．开源社区网[引用日期2012-09-25]