# 权限管理:

## 适用范围:中大型软件项目才需要 "权限管理"

## 权限管理:谁能做哪些操作,谁禁止做哪些操作

## (1)简单权限管理

需求

1 tom 普通管理员 查询数据(添加)

2 jerry 普通管理员 查询数据(添加)

3 kk 普通管理员 查询数据(添加)

4 james 系统管理员 添加 删除 查询 更新

5 dongdong 系统管理员 添加 删除 查询 更新

### (1)xz\_user 添加一列

uid uname upwd isdel type(INT)

1 tom 0 0

2 jerry 0 **0**

**3 kk 0 0**

4 james 0 1

5 dongdong 0 1

type:用户类型 0 普通用户(登录/查询/)

1 系统管理(登录/查询/删除/更新/添加)

### 1:登录模块(logindo.php login.html login.js)

#### 登陆成功：php将用户类型发送返回

#### 保存sessionStorage.setItem("type",0);

### 2:用户列表(userlist.php userlist.html userlist.js)

编号 用户名 电话

1 tom 123

var type = sessionStoarge.getItem("type");

if(type==1) {

删除 更新 详细

}

## (2)严格权限管理

### 概念:

RBAC基于角色的权限访问控制（Role-Based Access Control）

企业内部:

CRM客户关系管理/OA 办公自动化/ERP 企业资源管理

互联项目

商城

### 创建表:{5表基础}

#### (1)用户表 {保存用户信息}

#### (2)角色表 {保存角色信息}

角色:一组权限集合

角色1:10权限

角色2:2权限

#### (3)用户和角色关系表

#### (4)模块表(记录项目各个模块) 商品模块 用户名模块 订单...

#### (5)权限表(哪个角色对哪个模块有什么样的操作) CRUD

# 分页查询产品列表

## 1:SQL 复杂

xz\_laptop\_family 产品类别表 [name]

xz\_laptop 产品表 [title/price]

xz\_latop\_pic 产品图片 [sm]

### a:所有表名放FROM 后面

### b:给每张表起一个别名

### c:找不同表之间关系(列= 列> < <= ..)

二张表找一个关系;三张表找二个关系

a: FROM xz\_laptop\_family, xz\_laptop, xz\_latop\_pic

b: FROM xz\_laptop\_family f, xz\_laptop l, xz\_latop\_pic p

c: WHERE l.lid=p.laptop\_id AND l.family\_id = f.fid

SELECT f.fname,l.title,l.price,p.sm

FROM xz\_laptop\_family f, xz\_laptop l, xz\_laptop\_pic p

WHERE l.lid=p.laptop\_id AND l.family\_id = f.fid

GROUP BY l.lid

LIMIT 8,8;

## 2:PHP 当前页内容;当前页码;总页数

## 3:JS 分页[1][2]3[4][5]

# 删除商品

## 删除数据二种思路

### 1:在数据库直接删除该商品

DELETE FROM xz\_laptop WHERE lid = 1;

#### 问题:

(1) 删除表中记录数据库默认操作(在删除过程中锁定整个表)

不允许其它用户进行操作.

(2) 数据宝贵

## 2: 在数据库更新数据状态

xz\_laptop在表后添加一列 isdel 默认0 正常 1 补删除

UPDATE xz\_laptop SET isdel = 1 WHERE lid = 10;

### a:添加一列

ALTER TABLE xz\_laptop ADD isdel INT NOT NULL DEFAULT 0;

### 开发流程:

#### a:创建php data/product\_del.php

##### 1:获取参数 产品编号

##### 2:创建数据库连接/创建SQL语句/发送sql语句;

##### 3:返回 json {"code":1,"msg":"删除成功"}

#### b:分析: product\_list.html

#### c:创建js:完成删除操作

# 文件上传

功能:通用功能示例:上传头像,上传文件,邮件附件

图示:

文件上传到了哪里

## 开发方式:

### a:传统表单+ php (同步上传-图片不能预览查看图片)

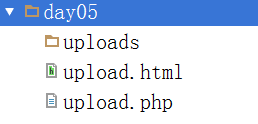
### b:表单+H5+AJAX+php (异步上传-图片预览图片)

## 开发流程(传统表单 + php)

### 1:在apache创建目录uploads(上传图片结果保存目录)

mac 报错 403 权限不足: chmod 777 uploads

### 2:创建一个表单(上传文件域/submit) upload.html



<form action="" method="post" enctype="multipart/form-data">

<input type="file" name="myFile" />

<inpu type="submit" value="上传" />

</form>

#### 1:method="post" 提交方式一定是post

#### 2:enctype="multipart/form-data" 加入特殊属性

#### 3:文件域 <input type="file" name="myFile" /> myFile 文件

#### 4:加入提交按钮

### 3:创建一个php(上传操作) upload.php

#### 1:php 全局数组 $\_FILES 保存上传文件信息与数据

$\_FILES["myFile"]["name"] 上传文件名称

$\_FILES["myFile"]["type"] 上传文件类型

$\_FILES["myFile"]["size"] 上传文件文件大小 单位字节

$\_FILES["myFile"]["tmp\_name"] 上传文件名称临时文件名称

$\_FILES["myFile"]["error"] 上传文件出错

#### 4：函数将临时文件移到指定目录下 uploads

move\_uploaded\_file(临时文件,uploads/文件名);

#### 5：上传图片名不能冲突

a.jpg---->3923828723.jpg

a.jpg---->2382728393.jpg

电脑容量单位

1 byte b[字节] 一个字母,一个数字,三分之一汉字

1kb = 1024b

1mb = 1024kb m兆

1gb = 1024mb g 吉

1tb = 1024gb t 梯

1pb = 1024tb p

## 方式二: 表单+H5+AJAX+php (拖动上传)

### 1:拖动文件上传介绍 Firefox/Chrome 直接从操作系统拖动文件，并上传服务器.

### 2:核心代码

#### 2.1:实现文件拖动效果,阻止浏览器默认行为,

四个事件(拖动离开;拖动释放;拖动进入;拖动)

$(document).on{drop:function(e.preventDefault())}

#### 2.2:HTML5通过拖动获取文件信息

e.dataTransfer.files

#### 2.3:实现预览图片(读取图片显示网页)

fileReader

webkitURL

var img = webkitURL.createObjectURL(files[0]); #兼容性好一些

var str = `<img src="${img}" />`;

$("#div").html(str);

#### 2.4:AJAX将图片上传

**var xhr = new XMLHttpRequest();  
 xhr.open("post","upload.php",true);  
 xhr.setRequestHeader("X-Requested-With",  
 "XMLHttpRequest");  
 var fd = new FormData();  
 fd.append("mypic",fileList[0]);  
 xhr.send(null);**