

# day1

## 品牌管理

### 需求

1. 品牌的增删改查
2. 品牌删除使用逻辑删除(status=>-1)
3. 图片上传使用uploadify插件,上传成功后回显图片
4. 图片上传到七牛云

### 数据表设计

brand 品牌

字段名	类型	注释
id	primaryKey	
name	varchar(50)	名称
intro	text	简介
logo	varchar(255)	LOGO图片
sort	int(11)	排序
status	int(2)	状态(-1删除 0隐藏 1正常)

### 插件

- composer安装uploadify "xj/yii2-uploadify-widget": "~2.0.0",
- composer安装七牛云 "crazyfd/yii2-qiniu": "dev-master",

### 知识点

- CDN
- AJAX上传文件
- 逻辑删除

# day2

## 文章分类管理

需求

- 1. 文章分类的增删改查

数据表设计

article\_category 文章

字段名	类型	注释
id	primaryKey	
name	varchar(50)	名称
intro	text	简介
sort	int(11)	排序
status	int(2)	状态(-1删除 0隐藏 1正常)
is_help	int(1)	类型

文章管理

需求

- 1. 文章表使用垂直分表,分两张表保存,文章表(article)和文章分类表(article\_category)
- 2. 文章表的增删改查(多模型同时输入)

数据表设计

article 文章

字段名	类型	注释
id	primaryKey	
name	varchar(50)	名称
intro	text	简介
article_category_id	int()	文章分类id
sort	int(11)	排序
status	int(2)	状态(-1删除 0隐藏 1正常)
create_time	int(11)	创建时间

article\_detail 文章详情

字段名	类型	注释
-----	----	----

article_id	primaryKey	文章id
content	text	简介

## 知识点

- 垂直分表和水平分表

一组讲解

## day3

## 商品分类管理

### 需求

- 商品分类增删改查
- 使用嵌套集合保存分类数据
- 使用ztree插件展示商品分类

### 数据表设计

article\_category 商品分类

字段名	类型	注释
id	primaryKey	
tree	int()	树id
lft	int()	左值
rgt	int()	右值
depth	int()	层级
name	varchar(50)	名称
parent_id	int()	上级分类id
intro	text()	简介

## 插件

- composer安装嵌套集合插件 "creocoder/yii2-nested-sets": "^0.9.0",
- ztree插件官网 <http://www.ztree.me>

# 知识点

- 嵌套集合

# day4

## 商品管理

### 需求

1. 商品增删改查
2. 商品相册图片添加,删除和列表展示
3. 商品列表页可以进行搜索
4. 新增商品自动生成sn,规则为年月日+今天的第几个商品,比如2016053000001
5. 商品详情使用ueditor文本编辑器
6. 记录每天创建商品数

### 数据表设计

goods\_day\_count 商品每日添加数

字段名	类型	注释
day	date	日期
count	int	商品数

goods 商品表

字段名	类型	注释
id	primaryKey	
name	varchar(20)	商品名称
sn	varchar(20)	货号
logo	varchar(255)	LOGO图片
goods_category_id	int	商品分类id
brand_id	int	品牌分类
market_price	decimal(10,2)	市场价格
shop_price	decimal(10, 2)	商品价格
stock	int	库存

字段名	类型	注释
is_on_sale	int(1)	是否在售(1在售 0下架)
status	inter(1)	状态(1正常 0回收站)
sort	int()	排序
create_time	int()	添加时间

goods\_intro 商品详情表

字段名	类型	注释
goods_id	int	商品id
content	text	商品描述

## 促销管理(选做)

---

### 需求

- 1. 促销信息增删改查

### 数据表设计

## 插件

---

- composer安装ueditor文本编辑器 "kucha/ueditor": "\*"

## day5

---

## 管理员

---

### 需求

- 1. 管理员增删改查
- 2. 管理员登录和注销
- 3. 基于cookie的自动登录
- 4. 管理员登陆后需要保存最后登录时间和最后登陆ip

## day6

---

## 项目讲解

---

1. 讲解自动登录
2. 讲解ueditor
3. 讲解商品搜索

## day7

---

### 需求

---

1. 使用RBAC管理用户和权限
2. 权限增删改查
3. 角色增删改查
4. 角色和权限关联
5. 用户和角色关联

## day8

---

### 需求

---

1. 菜单增删改查
2. 菜单和权限关联
3. 根据权限显示菜单

## day9

---

### 讲解

---

1. 四组讲解
2. 讲解菜单

## day10

---

### 用户模块

---

1. 用户注册

- 2. 用户登录（允许自动登录，登录成功记录最后登录时间和ip）
- 3. 收货地址管理

## 数据表设计

### 用户表member

字段名	类型	注释
id	primaryKey	
username	varchar(50)	用户名
auth_key	varchar(32)	
password_hash	varchar(100)	密码（密文）
email	varchar(100)	邮箱
tel	char(11)	电话
last_login_time	int	最后登录时间
last_login_ip	int	最后登录ip
status	int(1)	状态（1正常，0删除）
created_at	int	添加时间
updated_at	int	修改时间

## 插件

- 1. 阿里大于（发送手机短信）
- 2. 发送邮件

## 讲解

- 1. 避免短信接口被刷的解决方案
- 2. 避免网络不好验证码验证失败的解决方案

5组讲解当天模块功能

## day11

## 需求

- 1. 首页

2. 展示商品分类列表
3. 展示商品列表
4. 商品详情
5. 展示商品详情

晚上6组讲解

## day12 购物流程

---

讲解购物流程

- 购物车设计
- 参照主流的购物车设计方案(京东),完成购物车的数据存放方式的设计
- 如果没有登录就存放在cookie中
- 如果已经登录,就存放在数据表中
- 当用户登录的时候,将cookie中的数据自动同步到数据表中
- 如果已经有了这个商品,就使用cookie中的数量
- 如果没有这个商品,就添加这个商品到数据表

## day13 订单

---

- 在提交订单的时候:
- 判断库存是否足够
- 如果不够回滚
- 如果足够执行后续流程
- 执行成功清除购物车
- 晚上7组讲解模块内容

## day14 测试修复完善

---

## day15 全文索引

---