Day1：制作商品表，进行curd操作，搜索，排序

需求：商品表的操作：curd、搜索、排序、翻页

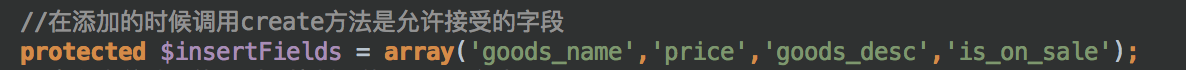
考虑到sql注入、xxs

使用js插件，时间插件，在线编辑

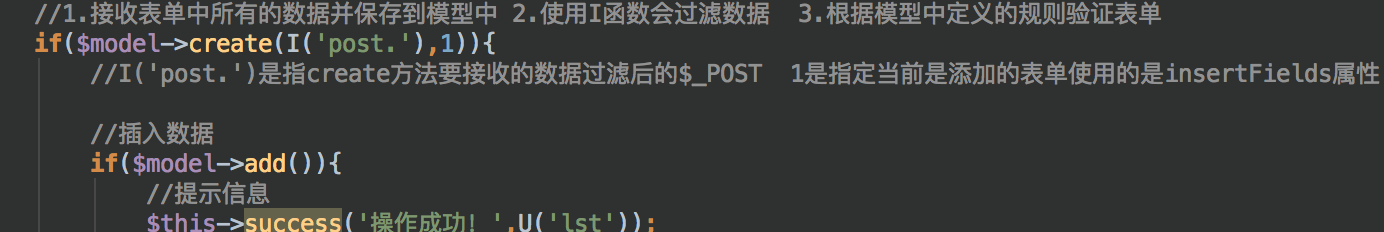
添加操作：存在一个安全问题：客户端可以假造表单（在表单中添加一个ID字段并设置了一个最大值，导致ID溢出无法再添加数据）。原因：我们在控制器中直接使用的create方法接收了表单不应该存在的字段。

解决方法：在Model中定义一个变量，在调用create方法时，只允许接收那些字段。

在模型中定义添加时允许的接收字段



在控制器接收是指定是添加操作



TP在调用crate接收表单时分三种情况：

1. 接收的是一个用于添加的表单
2. 接收的是一个用户修改的表单
3. 接收的是一个其他类型的表单

判断TP调用create方法时提交的当前是什么表单：

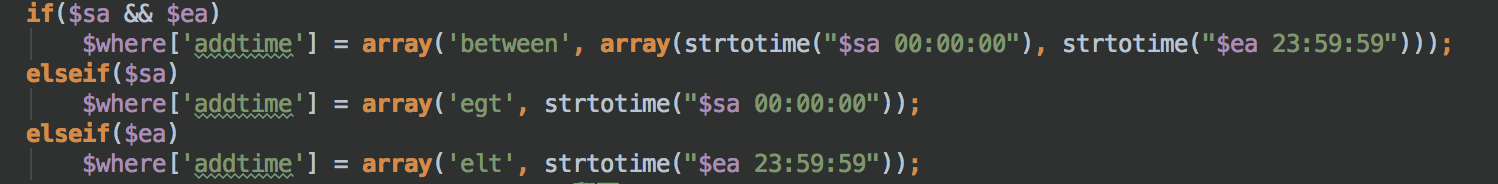
一：表单中有一个表的主键字段（id）就认为是修改

二：在create方法时传递第二个参数标记当前是一个什么类型的表单：1为添加，2为修改

缩略图：节省宽带，加载快

Day2：

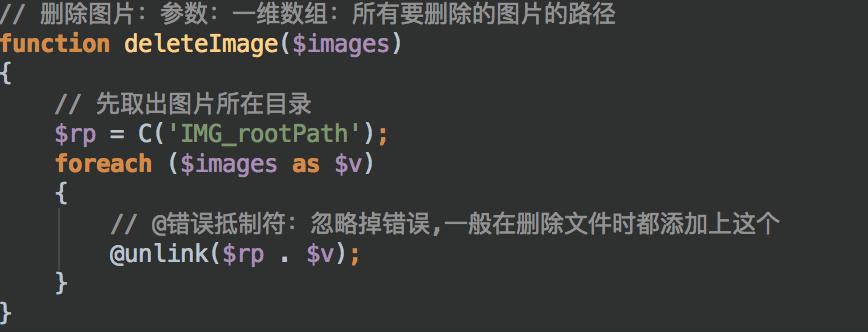
1. 使用时间插件完成添加时间的搜索功能（插件包datepicker放在Public目录下，修改模型中search方法添加代码）



1. 商品描述使用在线编辑器（插件包ueditor放在Public目录下）注意：在表单提交时在tp的调用了create方法，使用的I函数进行了过滤。默认的I函数会使用htmlspecialchars函数进行过滤的。如果在TP的配置中修改的话，容易被XSS（跨站脚本攻击：输入js代码或者html代码导致页面乱）攻击。这时候使用一个开源的过滤包：htmlpurifier包。创建一个函数，修改配置文件让I函数使用这个函数来过滤。



1. 商品删除要实现的功能：
2. 删除商品时从数据库中删除数据并且把图片从硬盘中删除掉
3. 在第几页删除的数据，就要在删除之后返回到第几页的用户体现。



1. edit：注意在修改的时候，要在修改前处理上传的图片，然后再删除原来的图片即可。
2. 添加后台模板样式。
3. 设置管理员登录。Index控制器的构造函数中添加验证的代码。让其他控制器都继承Index控制器。Admin不可以继承Index控制器，否则会出现只能登录之后才能登录的错误逻辑。
4. 封装自己的图片操作函数：Common/Common/function.php
5. RBAC操作：
   1. 权限管理（无限极，一个权限可以被多个角色拥有）
   2. 角色管理（可以吧权限分配给这个角色，一个角色可以拥有多个权限，一个角色可以有多个管理员）
   3. 管理员管理（可以指定管理员属于什么角色，一个管理员可以同时属于多个角色）
   4. 后台左侧只显示当前管理员权限的访问按钮
   5. 超级管理员（id=1）拥有全部页面，普通管理员只能访问分配了的权限的页面