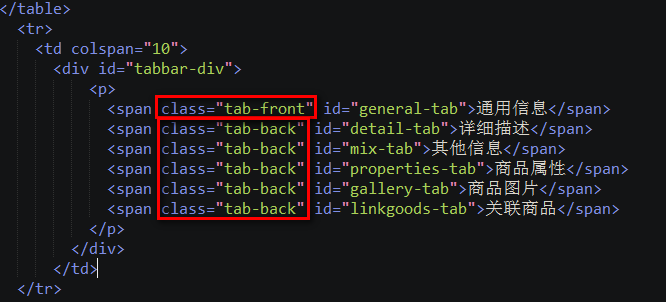
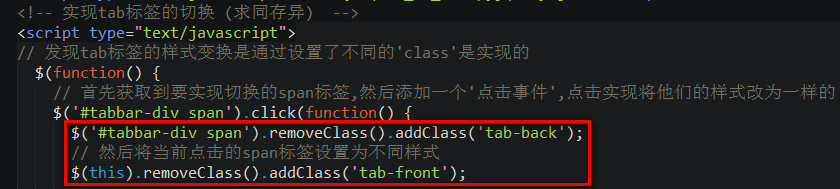
# 电商项目

1. 首先是模板的套入，就是用$this->display（）
2. 静态资源的引入，一般会在common下的config定义一下常量来代替，之后通过C方法来组装好路径，这样就不会出错。
3. D方法和M方法：D方法没有对应的模型的时候，实例化的是父类的模型。有对应的模型实例化相对应的模型。 M方法是实例化父类的模型。
4. 把一些有用的插件放在一个固定的目录下，建议是common下单建立个plugin
5. 引入富文本编辑器需要考虑一个问题，那就是xss攻击的问题，所以需要一个东西来解决这个xss攻击，本个项目中用的是一个插件叫做htmlpruifier插件来解决，这个插件的作用是过滤html标签，让什么标签可以通过，什么标签不可以。

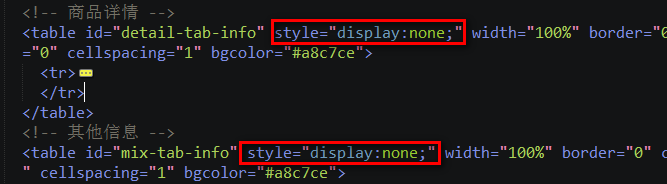
## 以下是后台商品处理的：

1. 要合理的利用别人的代码，ecshop中的样式代码，利用浏览器可以拷下来当成自己的代码。就是在添加那些样式。
2. 在实现标签的轮换效果时，体现的是求同存异的思想，先把第一除外的所有标签的样式都设为一样的(思想:相同的类名相同的样式属性)，当你动哪个就改变哪个样式。(每次点击时，将所有的span改成tab-back,再将当前对象$this改为tab-front)





1. 实现点击标签显示对应的区域而不显示其他的(一个标签对应一个tab)，体现的同样是求同存异的思想，先把第一个除外的所有table都设为隐藏(display:none) ，当点击哪个标签时(当前对象$this)就使其对应的table显示，其他的table隐藏。

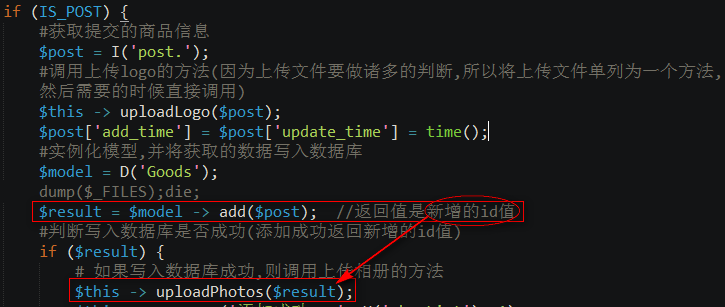




1. 商品logo上传主要是利用tp中的upload类和image类，一个是上传，一个是制作缩略图，上传时别忘了给form表单添加enctype=”multype/form-data”的属性，另外别忘了设置上传文件的保存的目录（自己创建），缩略图的保存的路径不能和上传的图的名字一样，需要自己平凑。
2. 商品相册实现：动态的实现相册的添加和删除主要用到的是jquery的append()和remove()方法，把table获取到，之后在他后面添加（append（想要的内容）），删除的是tr，所以需要先找到tr对象后再删除



1. 商品相册上传(多图上传)，首先需要定义一个变量($hasUpload:判断是否有图片上传)，之后根据判断处理图像，主要是循环处理图片(foreach)的上传和制作不同大小的缩略图(利用内置的upload类和image类)，特别注意: 我们上传相册图片时，并没有goods\_id,只有等商品入库以后，获取了goods\_id,再将相册信息入库



1. 最后商品修改，最重要的是在你点击跳转edit时一定要携带你想修改的商品id，用于查询数据并回显。(别忘了:添加隐藏域,用于post提交要修改的数据时,有商品的id值)

思路：商品的logo可以直接修改(详见下13); 相册的修改分两部分：图片删除通过ajax来实现单张图片的删除，图片添加则走原逻辑实现添加

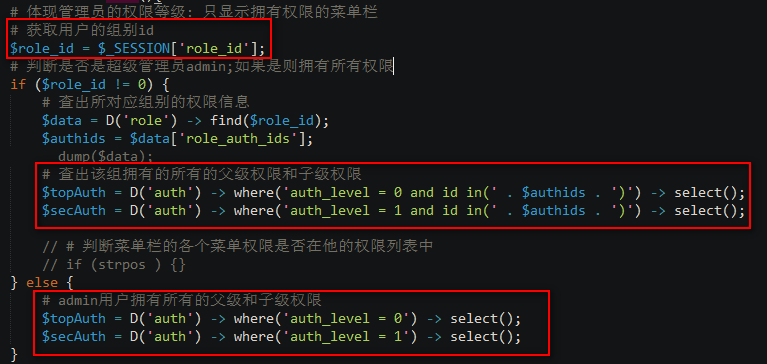
1. 商品logo修改：主要就是利用之前写好的那个上传logo的类（单独写的），只加一个判断条件，即判断是否是二次修改，如果是修改的话，提交的数据中肯定有隐藏域中的id，通过这个id值可以查出原来图片的位置和名字，删除原图(unlick()方法)，之后新上传的图片入库就行。
2. Ajax删除图片：使用的是post方法，把想删除的id传给处理的方法，根据返回值，移除图片所在的元素（这里面是移除span），方法中$this->ajaxReturn(array('status'=>1,"msg"=>'ok')); 这个就是返回数据的。

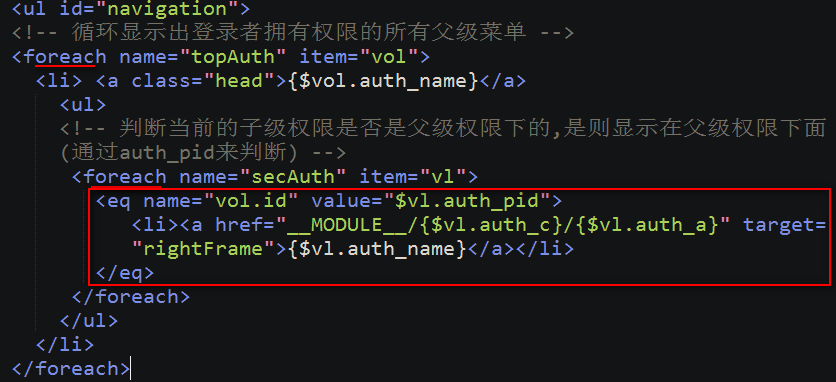
## 以下是权限控制的思路

1. 这次电商中的权限控制与上次tp中的权限控制有所不同，上次的权限控制是写在配置文件中的，而这次的是写在数据库中的，通过调用数据库数据来完成一些数据的显示。
2. 首先要有三张表：一张成员表（manager），一张权限表（auth），一张组别表（role）

一个成员对应一个组，一个组对应多个权限。

1. 首先需要改的是，index的left那个模块，登陆成功后,只显示拥有权限的菜单栏，而不是把所有的菜单全部显示出来(admin用户显示全部的菜单)
   1. 首先要获取的是登录者的’ role\_id ’
   2. 根据这个role\_id去查询属于哪个组，之后查询role表得到该组拥有的的权限id
   3. 之后利用得到的权限id查询auth表得到具体的权限，这里用到where条件子句，而且权限分为顶级权限和次级权限要分别将这些权限都查出来，这样在回显数据时便于处理，
   4. 注意的是超级admin是拥有所有的权限的，所以需要做一个判断，那就是你存入session中的uname是admin的话，就显示全部菜单（看iii）
   5. 这里面还用到了一个tp中以前没有学过的<eq name=’…’ value=’$...’>执行的代码段 </eq> name的值和value的值相比较，如果相等执行eq之间的代码，这里用来判断的是如果子级权限的pid和父级权限的id相等就让子级权限在父级权限下显示，注意一点name中不带$ ,value 带$.





1. 之后就是组别列表的处理了（role）
   1. 列表不说了，就是显示数据
   2. 权限的修改，
      1. 权限的修改前提是获得你的想修改权限的组别id(即role\_id)
      2. 根据的role\_id查询出组别对应的role\_auth\_ids(即拥有的权限id)，
      3. 之后利用权限id去查询auth表中具体对应的权限（用where条件查，查出pid为0和1的）,将查到的权限在模板中以多选的形式(checkbox)展示出来
      4. 显示完成后，就开始提交工作了，根据表单中的提交数据我们可以得到组别id(role\_id)、和权限id(role\_auth\_ids)，唯一没有的就是权限控制器和方法(role\_auth\_ac),所以需要想办法方法来得到这个。

思想: 我们可以通过权限id查询auth表来得到权限id所对应的控制器和方法(即auth\_a和auth\_c) ，然后拼凑字符串从而的到我们需要的role\_auth\_ac。

另写一个方法来获取我们要的控制器和方法: 需要传递一个参数，就是role\_auth\_ids，为了方便使用可以先将从表单中得到的数组形式的role\_auth\_ids转为字符串形式，之后根据这个查询出所有的子级权限(即auth\_level为1的)，之后将数据循环取出，拼凑成数据role\_auth\_ac那样样式。(之所以不用查出顶级权限是因为从数据库中可以得知顶级权限没有对应的控制器和方法)

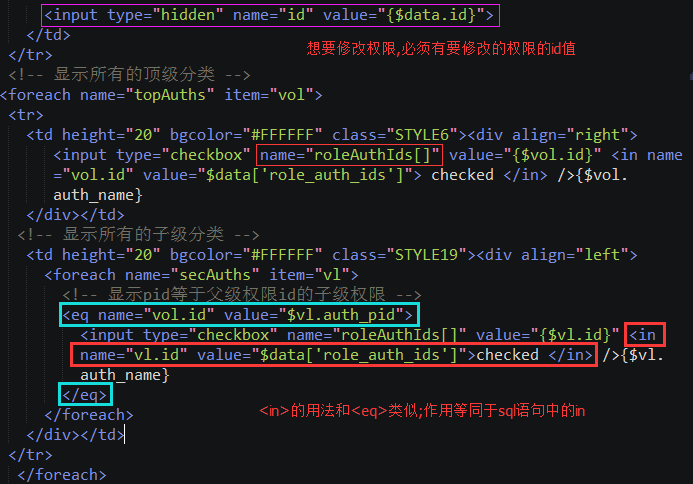
* + 1. 最后将的到的数据写入数据库更新就行了，（别忘了带id）



* + - 1. 只有在模板上的多选框中进行checked操作。

下面的查询用到了in语句，这个跟mysql中的in作用一样，表示在什么范围中，in 的value 是一个范围带$.

栗子中: 要求拥有的权限默认为被选中，所以只有id在这之中的才checked，





1. 之后是权限的显示和添加
   1. 显示不写
   2. 之后是添加权限
      1. 这里要做的是把权限列表显示在下拉框里，以便选择。
   3. 之后就是提交，刚开始提交的时候只能提交四个数据，缺少auth\_path和auth-level ,这就需要拼凑，也是一个方法专门来做这个的，这个函数需要传递两个参数，（注意的是这个函数是在添加成功后调用的属于更新操作，拿到id之后更新），这两个参数分别是添加id，和提交的数据。
   4. 之后拿到传过来的id 查询出他的父id，之后拿着父id 找到他的auth\_path，之后用这个auth\_path和data中的你提交的auth\_a拼凑成最后auth\_path，而auth\_level就是父的auth\_level+1就行了。