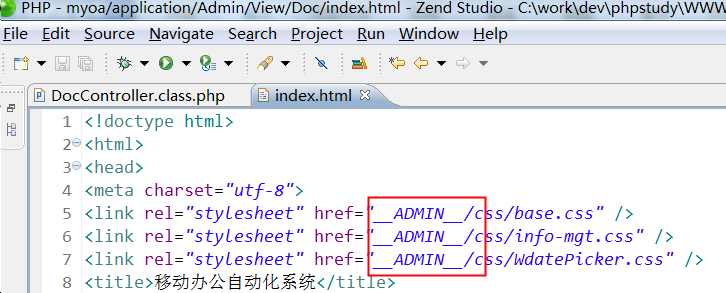
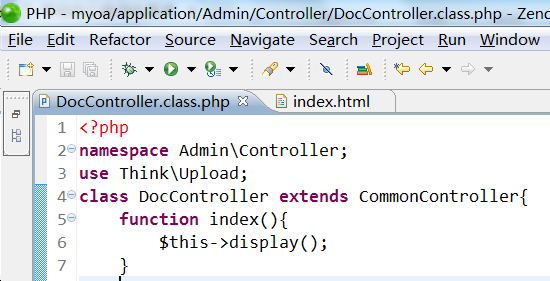
ThinkPHP框架-6

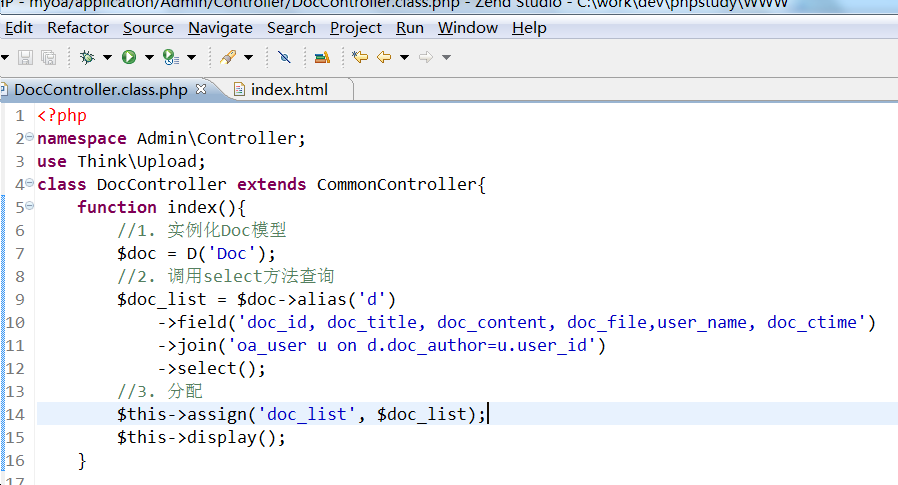
# 1. 综合案例--公文起草模块实现

## 1.1 公文列表

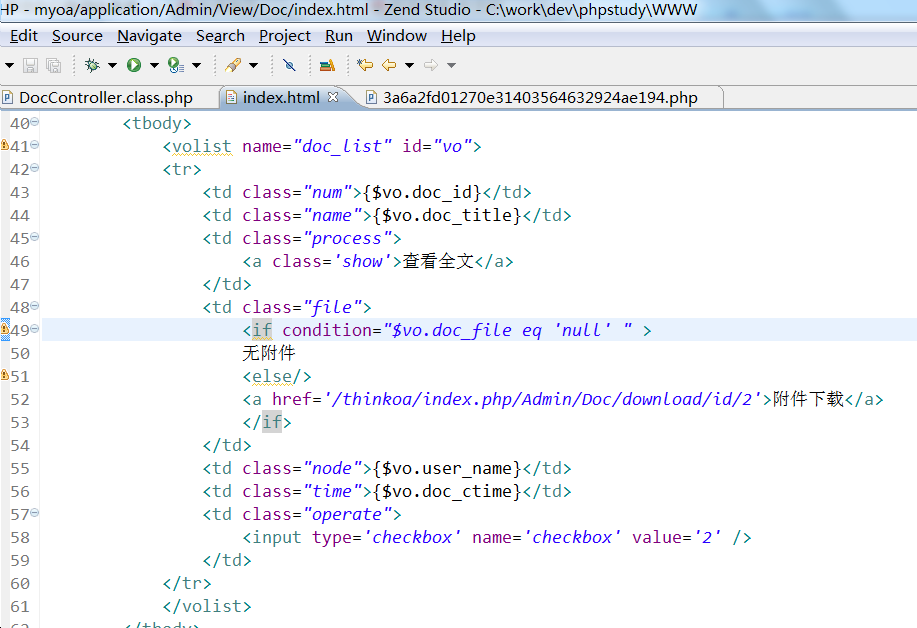
1. 创建Doc/index方法，调用视图，让视图正常显示



1. 修改Doc/index方法，从Doc数据表中获取数据



1. 修改视图显示



访问结果：



## 1.2 附件下载功能实现

PHP提供的4句话：

header("Content-type: application/octet-stream");

header('Content-Disposition: attachment; filename="' . basename($path) . '"');

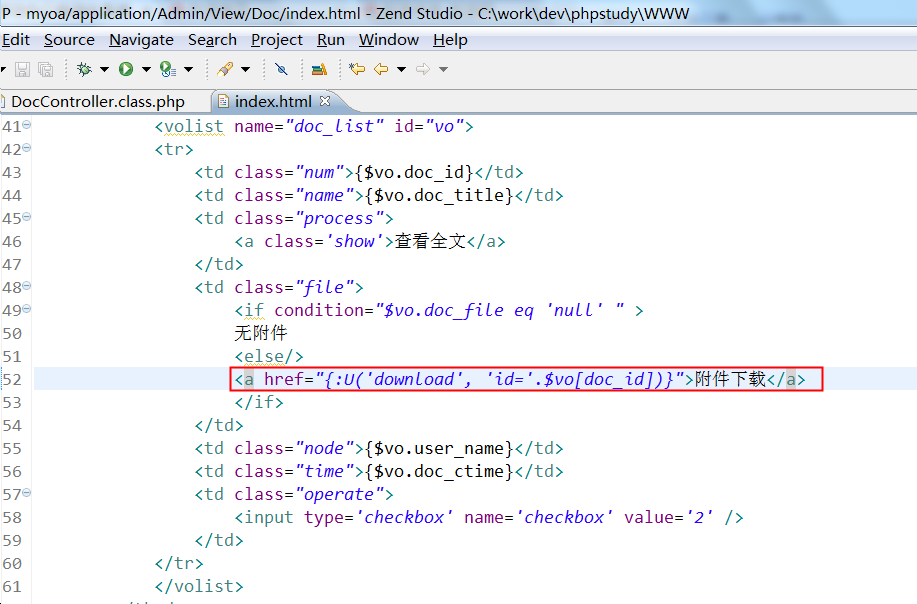
header("Content-Length: ". filesize($path));

readfile($path);

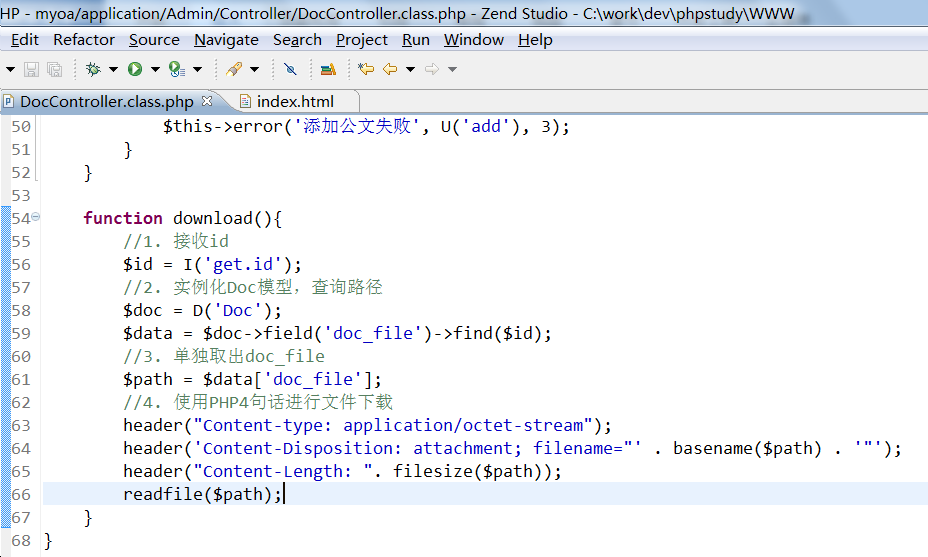
Doc\_file: ./Upload/2016-12-23/13298437885.png

思路： 附件下载链接，将公文id传给后台PHP控制器，控制器根据id查询路径，再进行下载

1. 修改Doc/index.html视图



1. 创建Doc/download方法，接收id，根据id查询路径，再进行下载

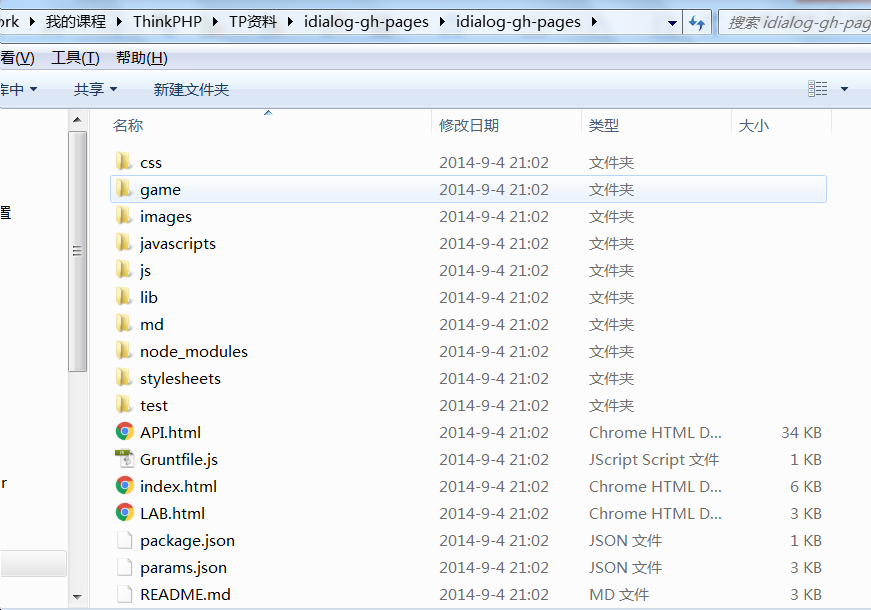


测试：能够正常下载各种文件

## 1.3 使用iDialog插件显示公文正文

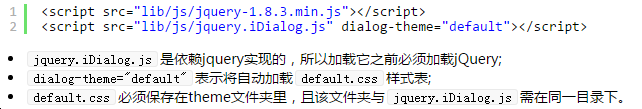


解压：

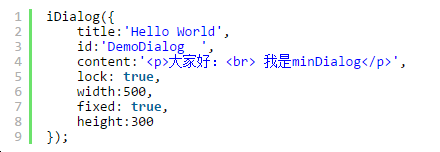


分析：

① 引入文件：



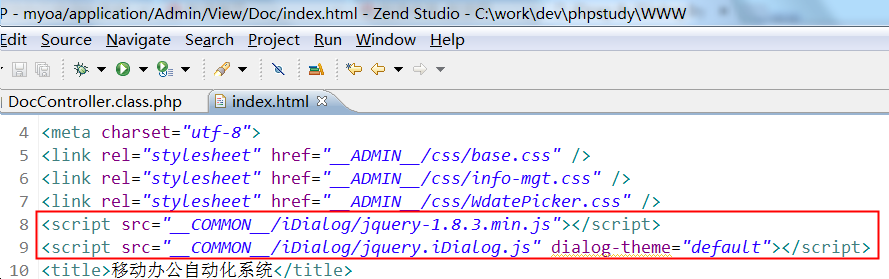
② 弹出框代码：



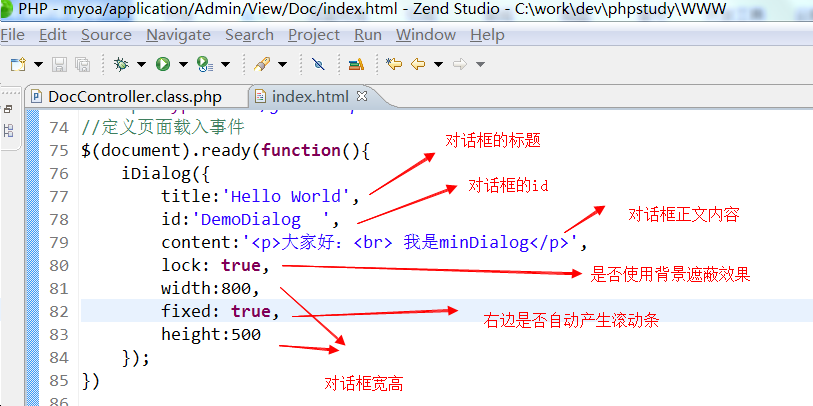
1. 将iDialog核心包复制到项目中



1. 在Doc/index.html视图文件中引入iDialog插件



1. 复制弹出框代码到index.html视图中

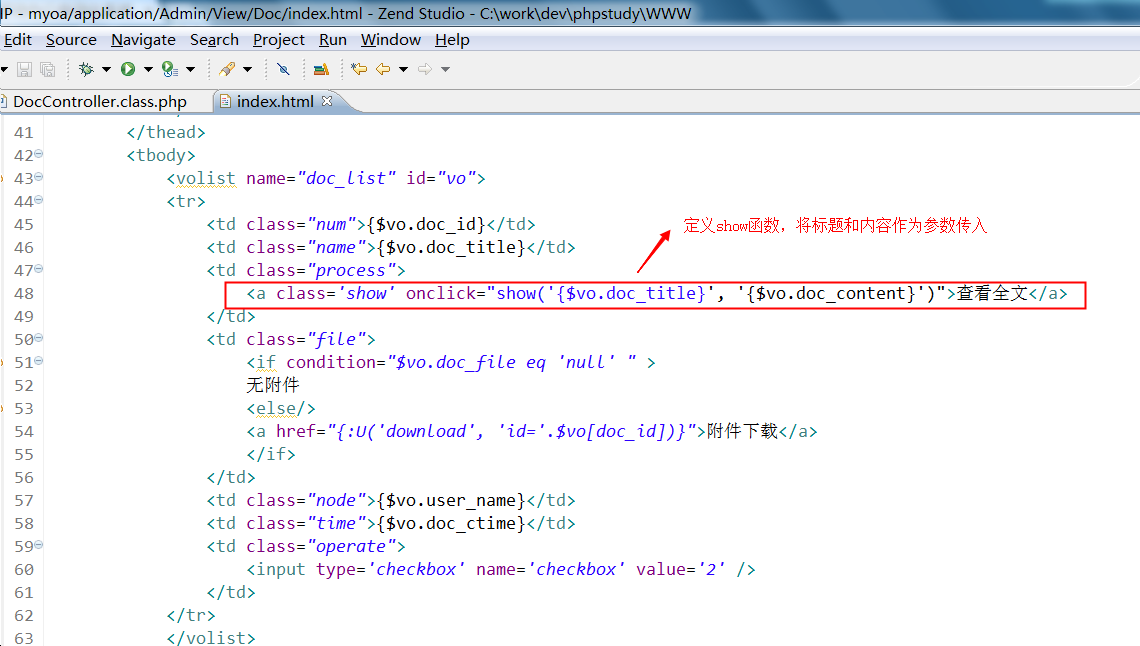


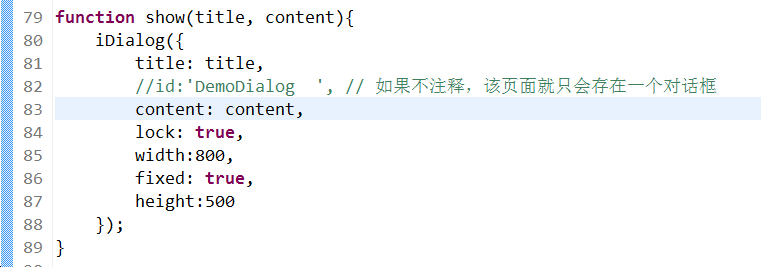
1. 点击“查看全文”，能够在对话框中显示正文内容

思路： 在“查看全文”按钮上注册点击事件，该事件能够弹出对话框，并显示内容

具体获取内容方式：

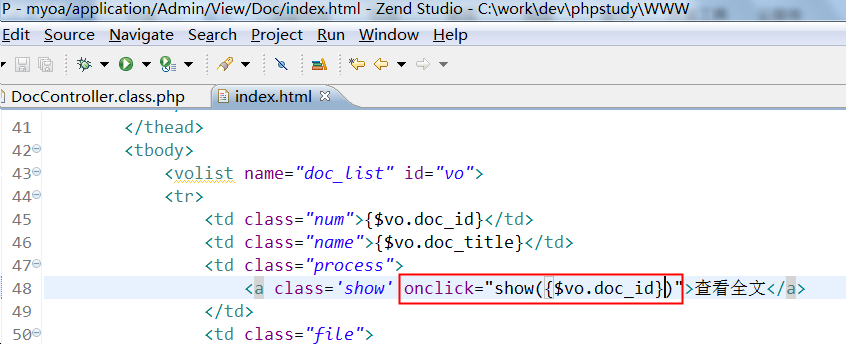
① 将标题和内容作为show函数参数传入

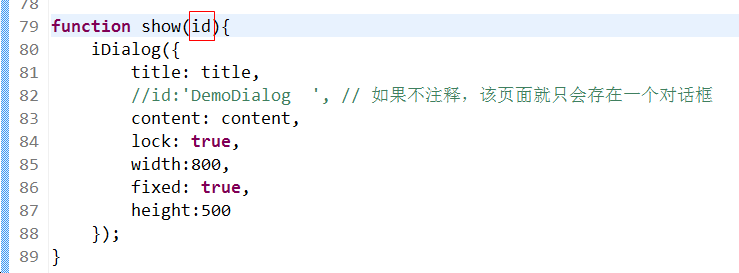




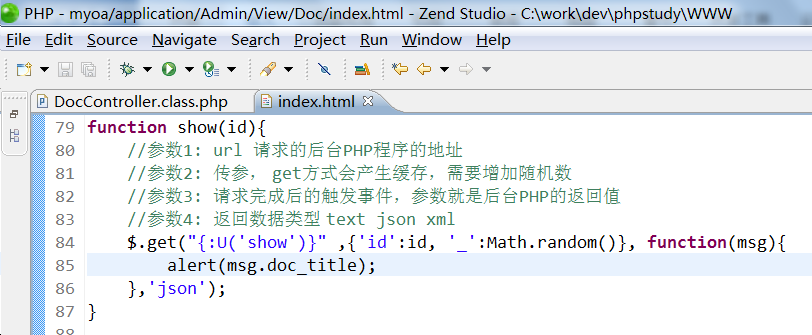
② 在show函数中，之传入doc\_id，调用ajax方法将doc\_id传入后台PHP程序，PHP程序根据doc\_id查询标题和内容，再返回给前台。前台再进行显示

I. 在show函数中，传入doc\_id

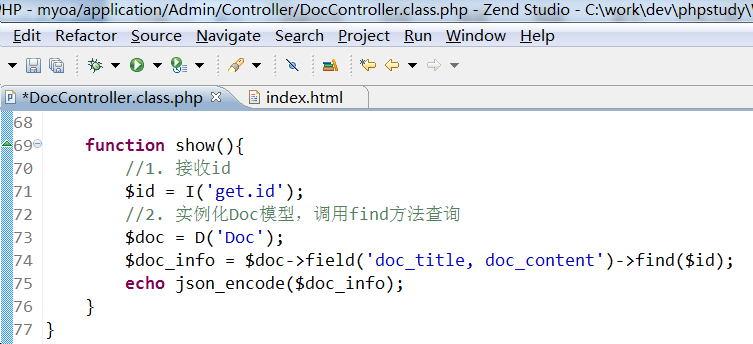




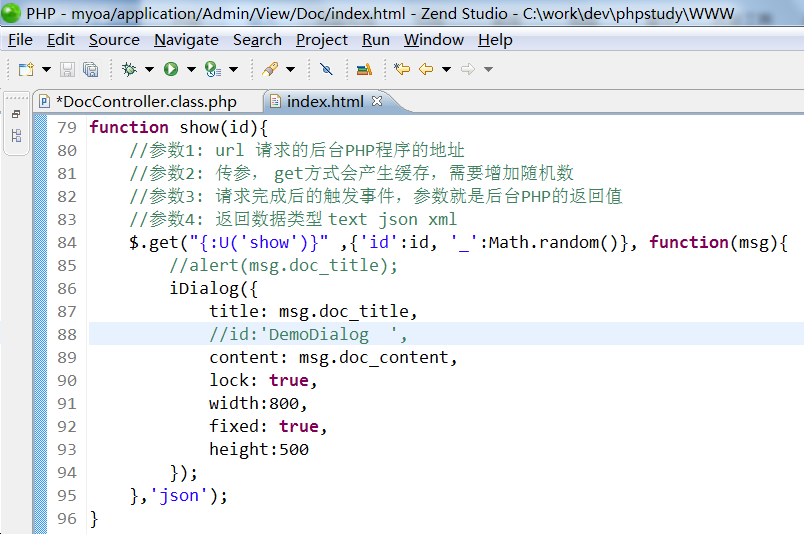
Ii. 修改show函数，是用$.get方法请求后台PHP程序



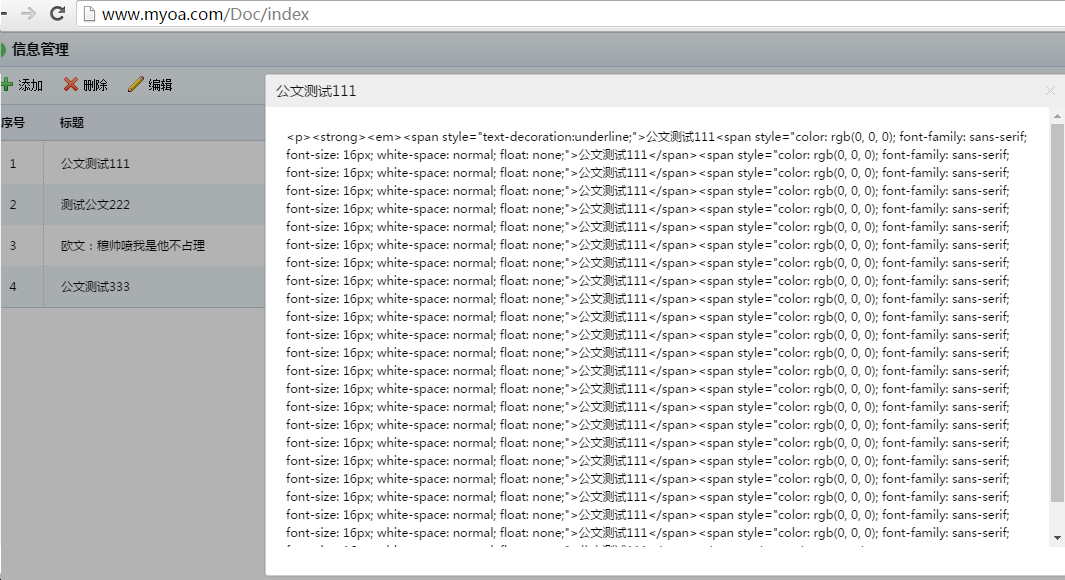
Iii. 创建Doc/show方法，根据接收到的id查询需要的数据



Iv. 修改index.html视图中的show函数，调用弹出框，显示内容

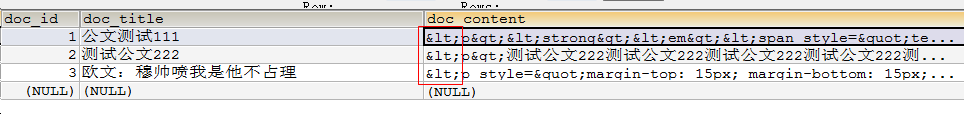


测试：

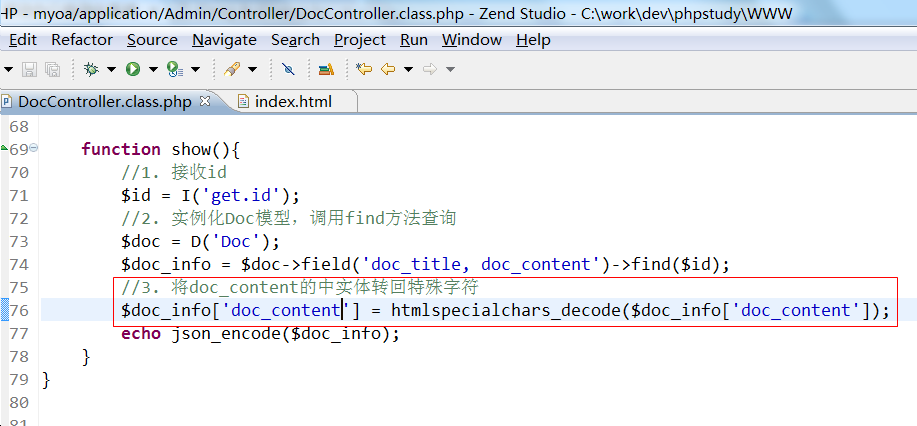


在网页上将标签原样显示了。

原因：在数据表中是以HTML实体形式存储的



V. 完善---Doc/show方法中，将HTML实体转回标签



测试：

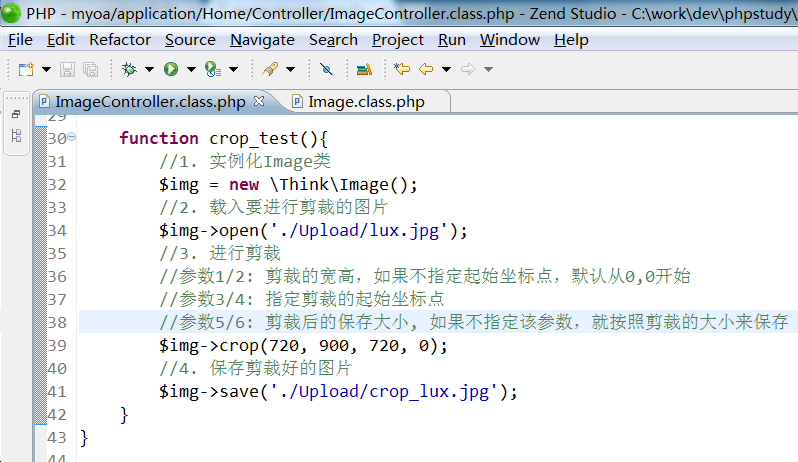


# ThinkPHP中的图像处理类

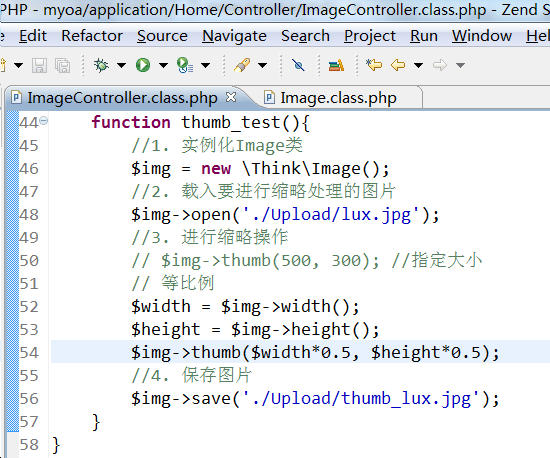
位置：ThinkPHP/library/Think/Image.class.php

操作： 水印、缩略图、剪裁

案例1: 剪裁

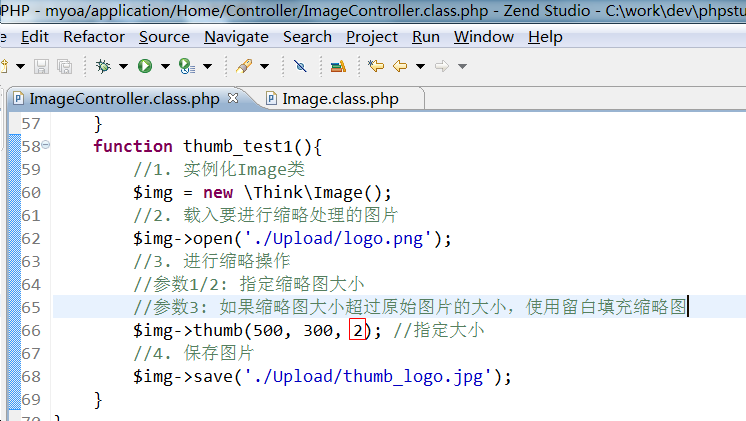


案例2：缩略图

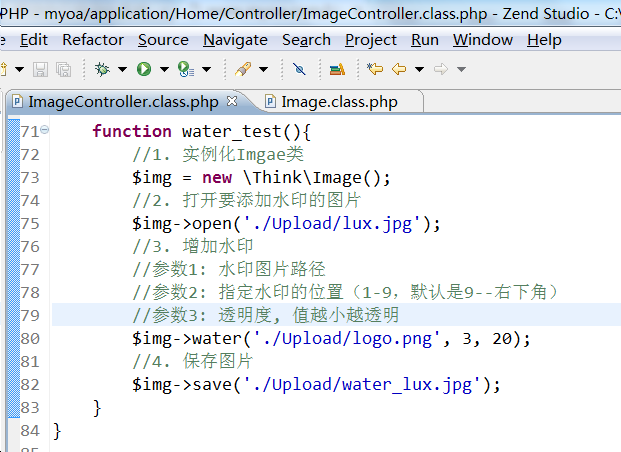


问题：指定缩略图大小时，指定的宽高超过原始图片的宽高，如何处理？

处理方式之一：留白



案例3： 水印--water(图片水印)、text(文本水印)



图像处理类内部：

TP框架同时支持 GD库和ImagICK库来进行图像绘制。

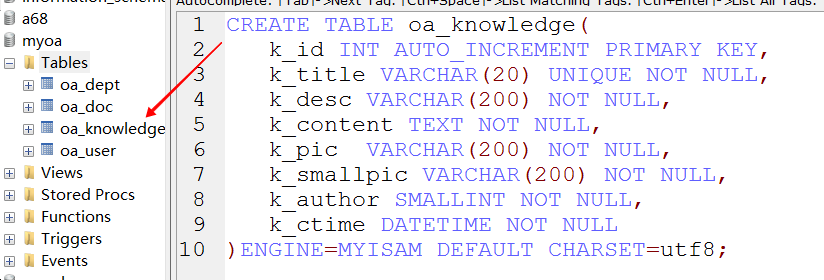
问题：具体使用的是哪种图像处理库

查看构造方法：



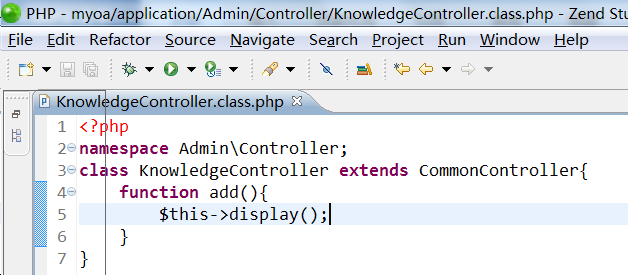
思想：驱动机制。 能够让程序具备更强的扩展性和通用性。

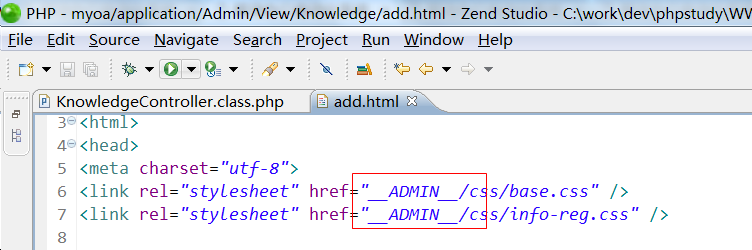
# 知识管理模块实现



## 3.1 添加知识--上传图像、制作缩略图

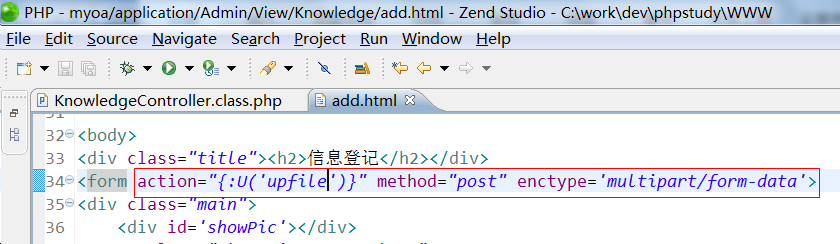
1. 创建Knowledge控制器和add方法，调用视图，让视图正常显示



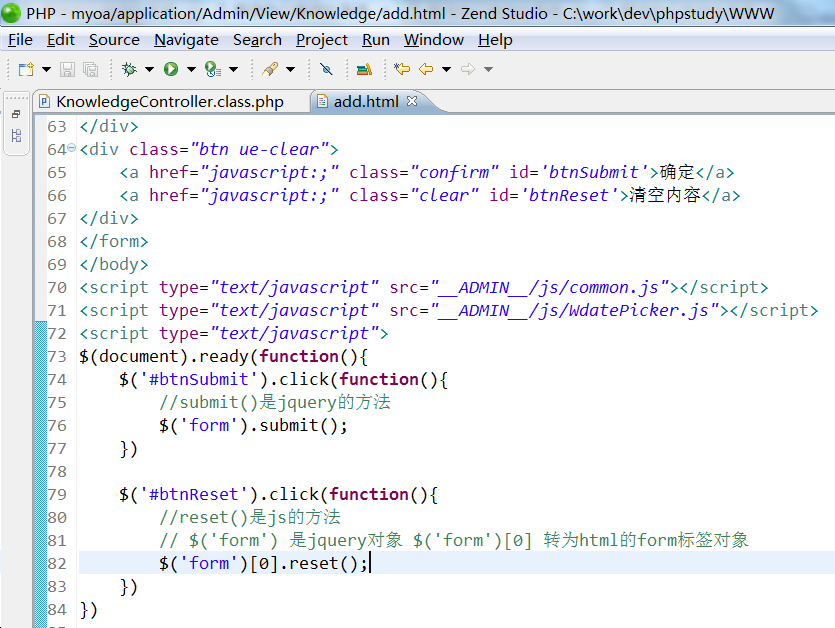


1. 上传图片，制作缩略图

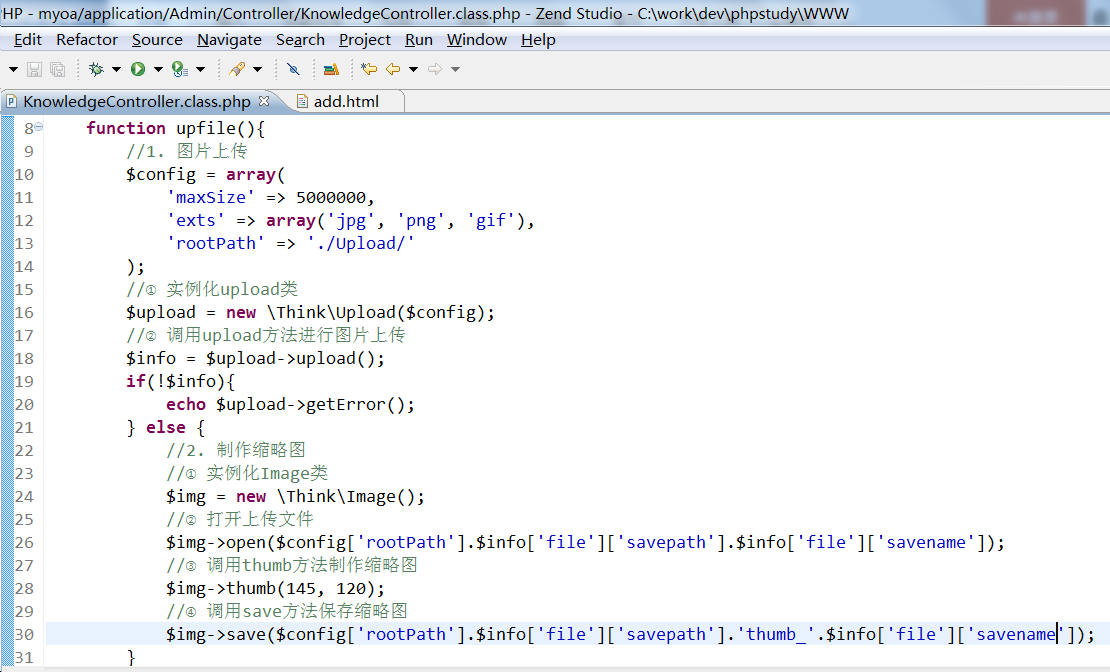
① 完善表单







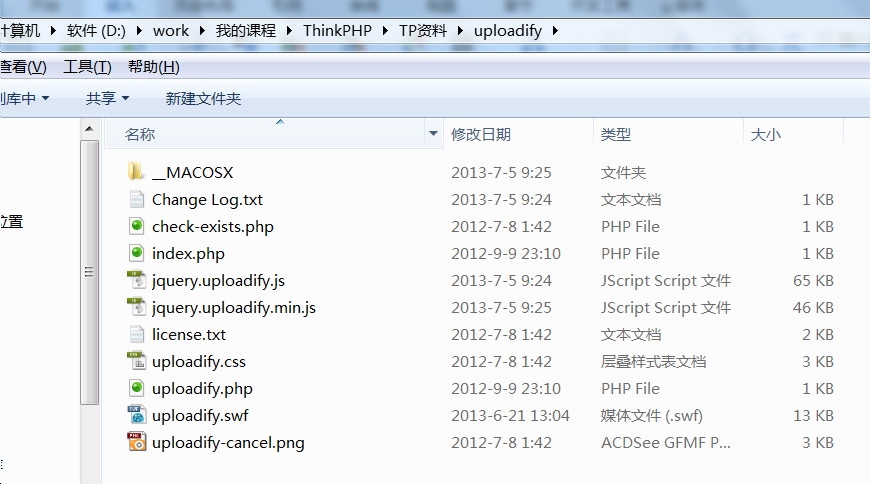
② 创建Knowledge/upfile方法，接收上传图片，制作缩略图



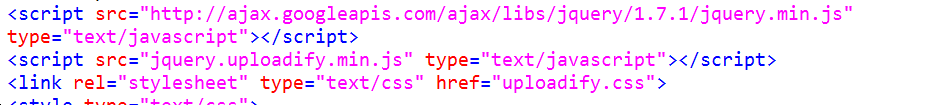
## 3.2 使用uploadify文件上传插件

基于jquery的一个ajax文件上传插件，并且还使用了flash。

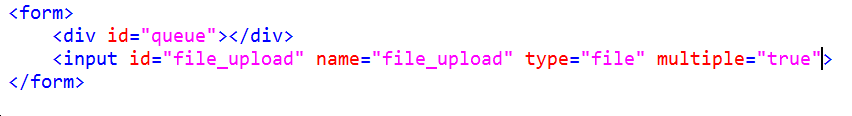




1）分析demo  
① 引入2个js文件和1个css文件

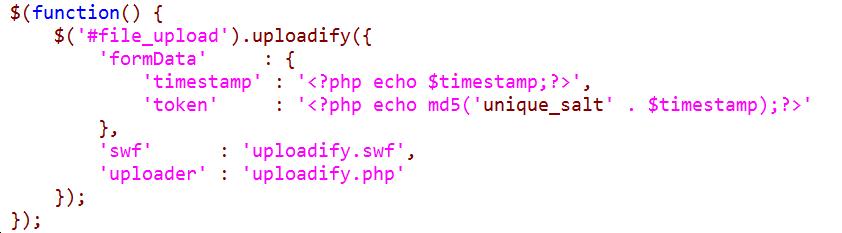


② 设置文件域



注意： 文件域中需要指定 multiple=’true’

③ 获取文件域标签，调用uploadify插件函数



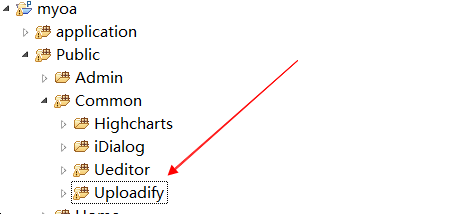
formData: 提交到后台PHP程序的数据

swf: 文件域使用flash文件

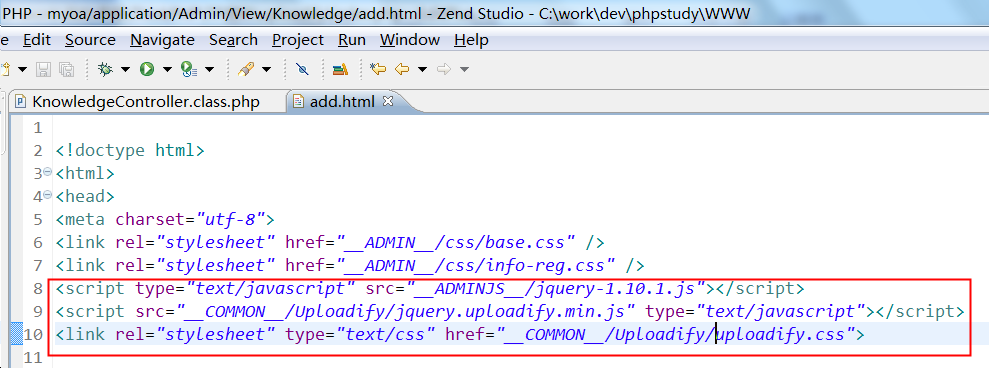
uploader: 后台进行文件上传的PHP程序

1. 将Uploadify插件加入OA项目

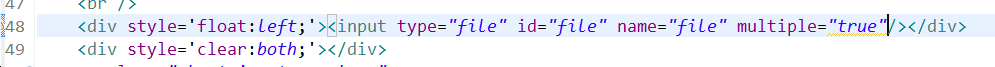
① 将Uploadify插件复制到项目中



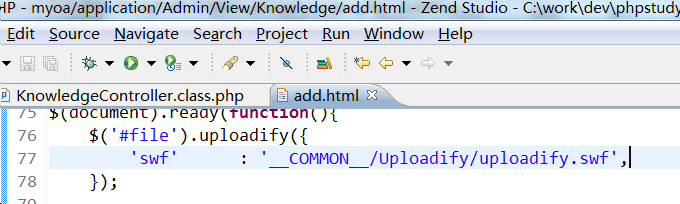
② 在Knowledge/add.html视图中引入css和js文件



③ 指定文件域的multiple="true"参数



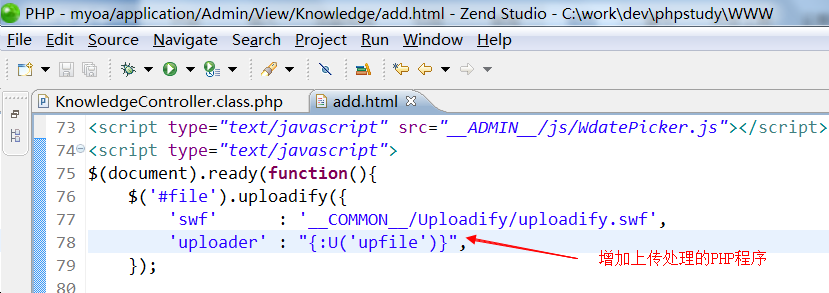
④ 获取文件域标签，调用Uploadify函数



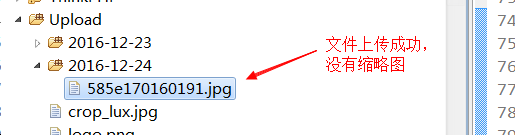
访问结果：



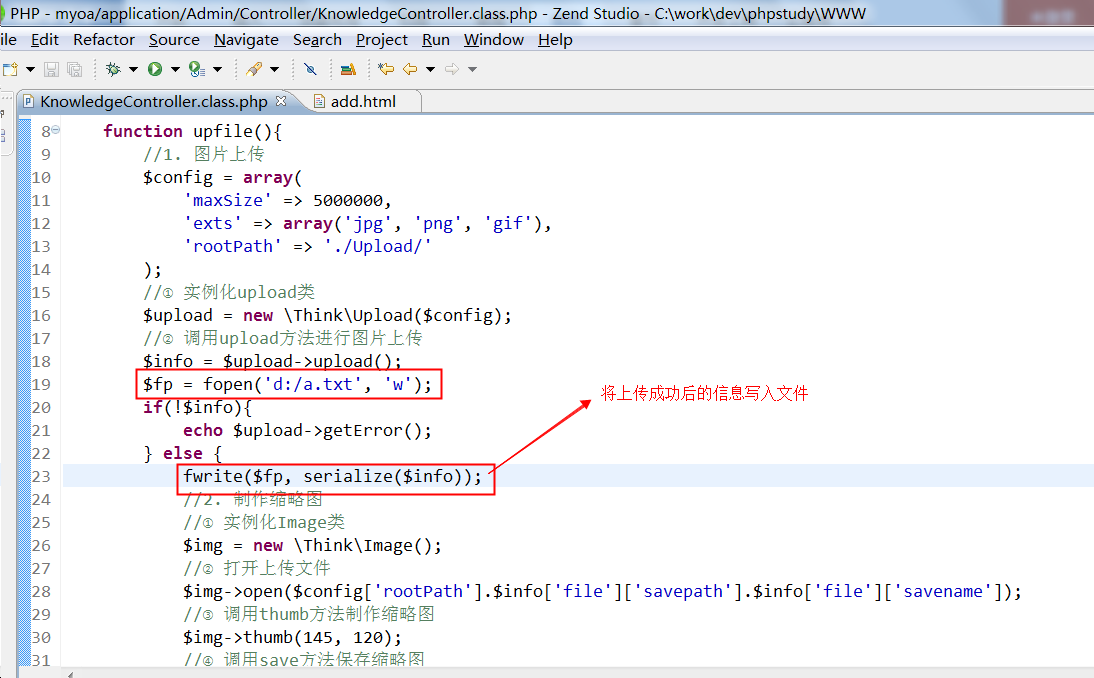
⑤ 定义uploader配置项进行图片上传，制作缩略图



测试：

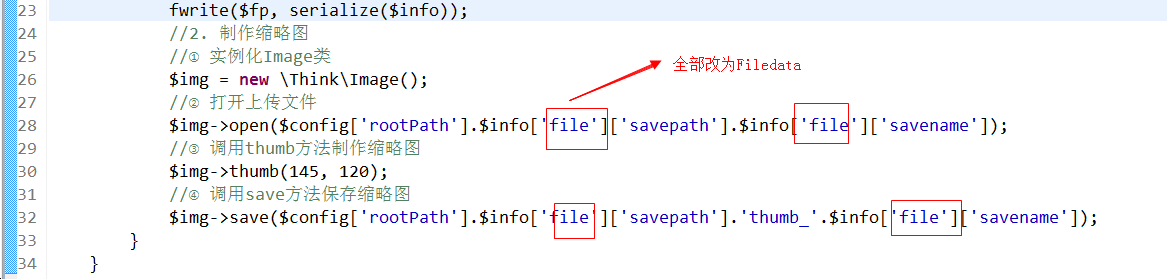


分析原因：通过写文件方式





所以：在制作缩略图时，需要修改路径参数

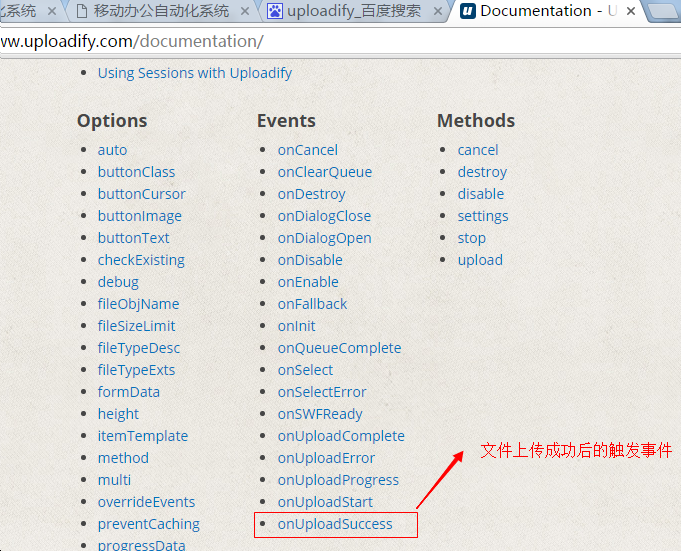


修改完成后再次测试：上传成功并且缩略图也成功。

Debug小技巧： 只要是文件处理时出现的错误，十之八九都是路径问题。

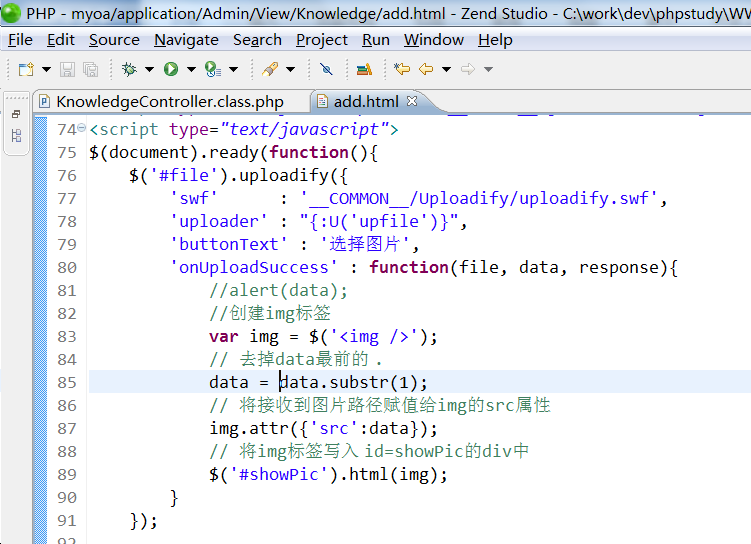
⑥ 在add.html页面上显示缩略图

查看手册：





在配置项中增减 onUploadSuccess触发事件



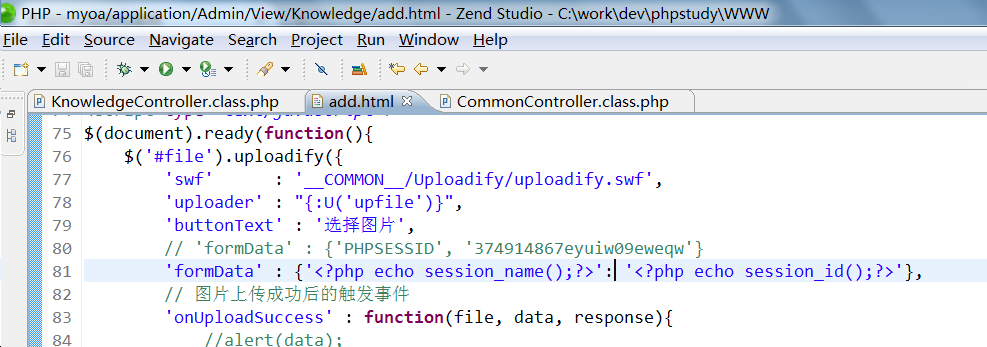
后台upfile程序，进行缩略图地址输出



Uploadify的bug： 和除了chrome浏览器结合之外，全都有一个session丢失的bug。

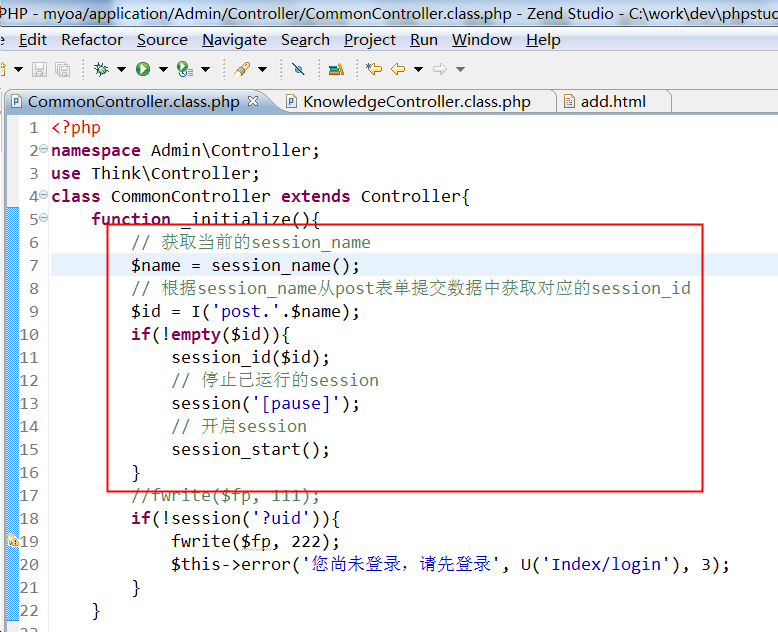
修复bug：

① 在add.html页面中将session\_name和session\_id都提交给后台程序



formData： 是以post方式提交数据的

② 修改CommonController中\_initialize()方法，重建session



在Firefox下测试，能够正常上传图片并制作缩略图

# 4. 邮件管理模块实现（站内信）

## 4.1 数据表设计

## 4.2 写邮件