原视频博客中没有对评论回复的功能，只能评论，不能对评论再进行回复，感受不太好，因此，做了这个修改。

主要的做法是按照bootstrap中媒体对象的写法，在回复中加入了缩略图，每个评论或评论回复后都有“回复”的链接，设计了js的点击事件，当用户点击“回复”时，会出现一个js的输入框和一个按钮，当用户在输入框中输完内容，点击按钮后，又会触发一个js事件，这个事件会做以下几件事：

1. 获取“回复”所在元素（不太准确）的用户名，以便在回复内容前加上“回复@XXX”,采用的方法是遍历评论及回复时事先把评论或回复的作者名存入到回复所在a标签的title属性中，然后：

**if**(**aA**[window.index].className=="comment"){  
 **var** content=' **<?php echo** '回复@'; **?>**'+**aA**[window.index].title +': '+inp.value;  
 }**else**{  
 content=' **<?php echo** '回复@'; **?>**'+**aA**[window.index].title +': '+inp.value;  
 }

1. 获取评论内容，这个比较简单；
2. 获取reply\_id，即我们新创建的评论或回复所针对的id,实际也就是用户点击回复时，回复所在的评论或回复的id（当然这里无论评论或评论回复都在comment表里，区分两者唯一的标准是所有的评论其reply\_id均为0，评论是针对文章的；而所有的评论回复的reply\_id都不为0，其值就是它所针对的评论的id，这里还做了一个小的处理，如果某个评论回复是对回复(假定用R代表)的回复而不是对评论的回复，那么它的reply\_id仍然指向上一层回复R所针对的评论的id。这样做的目的或者结果是使在一个评论下，不管是回复还是对回复的回复，都作为一层放在该评论下，不再区分是第几层回复，具体针对哪个作者我们已经使用了前面的“回复@XXX”来指明）；
3. 最后这一版，采用了ajax传输，为什么要用ajax呢? 我们在上一版使用了window.location方法从js中向控制器转向和传递参数，在运行完成刷新页面后，我们在js输入框内输入的内容每次都会出现在最后的评论输入框(\_guest页面)中，采取了一些办法，比如设置autocomplete=off等都不起作用，进一步发现，这和浏览器显示的url地址有关，如刚开始进入页面时的url地址就是正常的。但是转向语句如果用刚开始进入页面时的url地址的话就无法传输数据了，那么我们可不可以先用post方法传递数据，再传递完成后再用转向语句（这时可使用刚开始进入页面时的url地址）来刷新页面呢？根据以上想法，测试了一下，没有问题，只是在这个url地址中再加一个参数，其值为一个随机数，这样避免因缓存而不刷新页面。中间使用window.location出现一点问题，后使用location.replace强刷，没问题。最后发现window.location也没问题。

\*\*后注：window.location偶尔有不刷新情况。location.replace测试也偶尔有，后面再仔细测试location.replace就没了，？？？，原因不详。但这里又找了一个location.reload()，刷新当前页面，发现挺好，这个在刷新过程中不像前两个有页面跳转的过程的略微停顿，页面基本不动，稍等一下，新的评论就出来了，挺好。

1. 最后一版，为测试方便，在存入数据库时，将status全部设置为2，实际应用时如果需要审核，应设置为1，可在post/detail的step2里进行修改。
2. 如前所述，所有评论回复的数据都放在了comment表里(原来想单做一个表，后来觉得没必要)，对comment表增加了一列reply\_id, reply\_id为0就是评论（其实评论就是针对文章的回复），reply\_id不为0则表明该数据是评论回复（或简称回复），此时的reply\_id值就是回复针对的评论的id(comment表中的id)或者往上追述回复所针对的评论的id。
3. 由于修改了comment表，因此对commen类以及前台的comment的index页面稍做了修改。