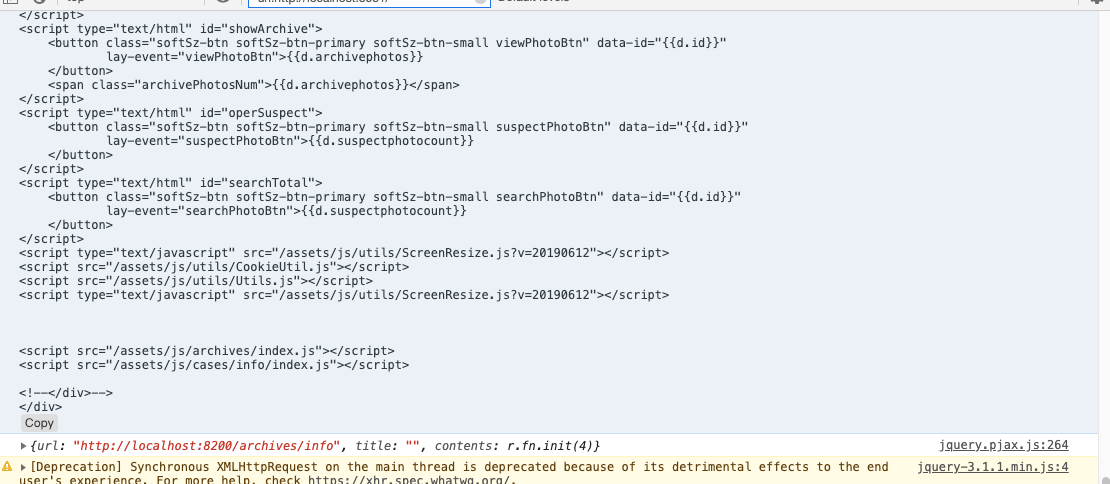
pjax加载页面不执行js插件的问题

Ps:  
1.每次服务器返回页面会调用**function**pjax(options) {}的options.success= **function**(data, status, xhr) {}页面然后会通过cachePush方法去把cacheMapping[id] = value更新  
2.每次浏览器前进后退时都会调用：  
**function**onPjaxPopstate(event) {}*/\*\*每次浏览器前进后退都会执行\*\*/,然后从***var**cache = cacheMapping[state.id] || []  
**var**containerSelector = cache[0] || state.container;//cache[0]是pjax容器的$对象  
// var container = $(containerSelector), contents = cache[1];////cache[0]是页面文本  
**var**container = $(containerSelector),  
contents = cache[1];  
  
**if**(container.length) {  
**if**(previousState) {  
// Cache current container before replacement and inform the  
// cache which direction the history shifted.  
// cachePop(direction, previousState.id, [containerSelector, cloneContents(container)])  
*/\*\*修改源码直接读取context.html()\*\*/*console.log(container.html());  
cachePop(direction, previousState.id, [containerSelector, container.html()]);//改成html就能被js监听到  
  
}  
 然后通过cachePop再次更新全局的cacheMapping[id] = value（保留了。id:[#pjax,"页面文本"]）  
每次跳转到新页面，都会把自己cacheMapping的缓存删除

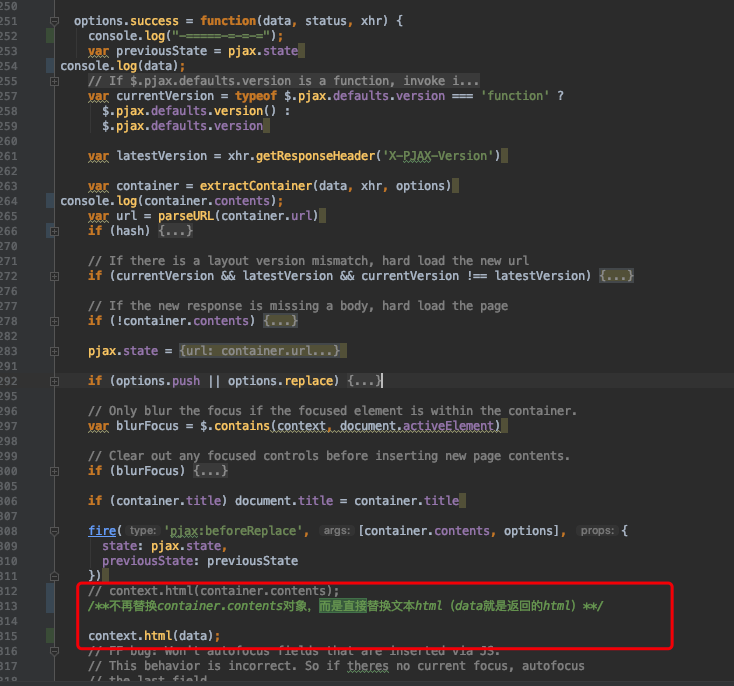
### 介绍

在使用jquery.pjax的时候发现加载页面时不会执行其中的layui以及jquery的初始化方法，包括一些插件的初始化方法。

查看源码后发现该jquery.pjax替换容器内容时，是将服务器端返回的html转换为了jquery dom对象然后再执行的替换，  
如图jquery dom对象container.contents  
  
  
  
  
但是这样操作会导致一系列的加载事件不会被触发，导致例如jquery和layui的初始化方法不会被执行。

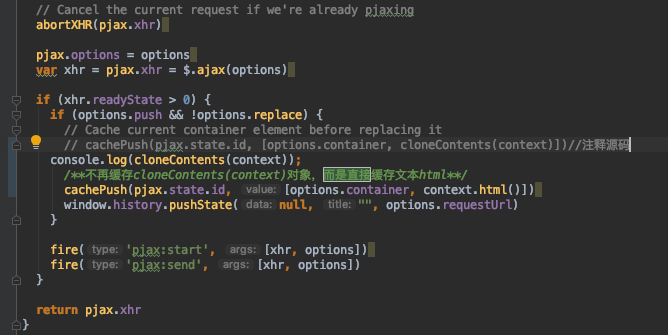
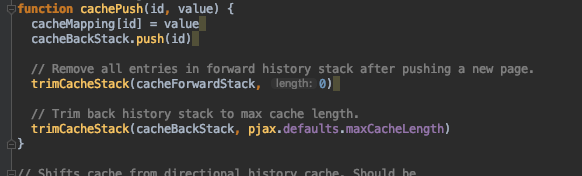
于是决定将添加dom节点修改为直接添加html片段（即是返回的data。局部替换context.html(data)  


### 修改代码

1.将jquery.pjax.js中311行的 context.html(container.contents)修改为 context.html(data)。  


    这样在pjax加载新页面的时候便会直接将服务器端返回html片段添加进容器。此时使用pjax加载新页面的时候，layui jquery等插件的初始化方法即可正确执行。

2.  但是这样仅处理了新增页面，而执行回退操作时pjax会从缓存中读取上一个页面的内容，同样pjax在回退上一个页面的时候依然是以dom对象键值对的方式存储和添加的，所以我们还需要继续修改。  
将365行的cachePush(pjax.state.id, [options.container, cloneContents(context)])修改为 cachePush(pjax.state.id, [options.container, context.html()])

  
  
  
3.按上面改完后发现。第一次前进后退已经正常了，然后第二次前进后退的时候依然会出现页面js失效的情况：  
所以还需要更改源码467行：  
 // cachePop(direction, previousState.id, [containerSelector, cloneContents(container)])

*/\*\*修改源码直接读取context.html()\*\*/*console.log(container.html());  
cachePop(direction, previousState.id, [containerSelector, container.html()]);//改成html就能被js监听到  
  
