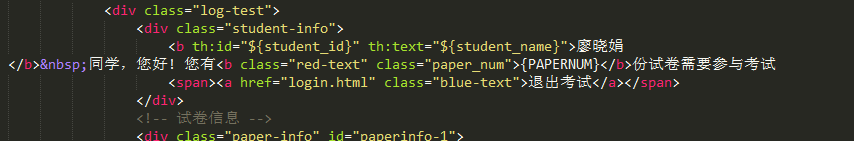
# 学生部分

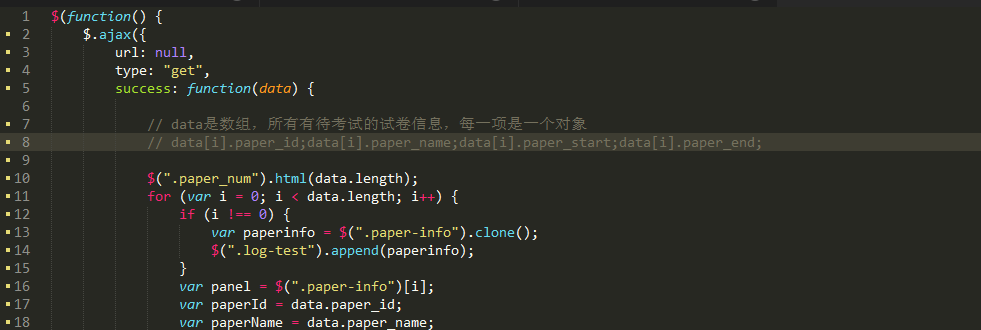
**Select.js**

**进入初始化服务器端需要保存登录信息，登录用户student\_name和id**

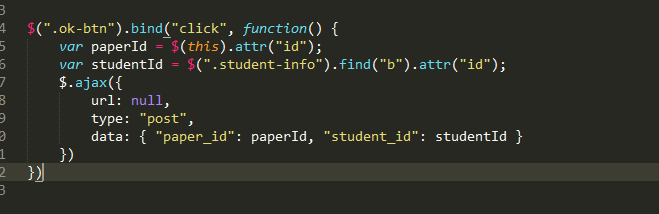


**页面加载完成后，浏览器发起请求，请补充请求url**

**返回数据如下：**

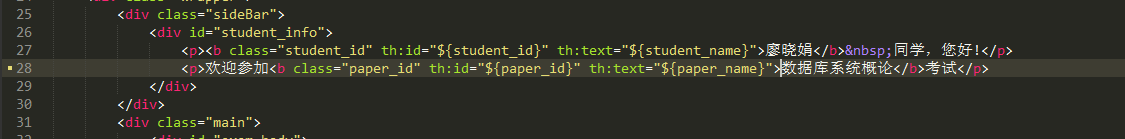


**学生选择完成后，会提交选择的试卷id，然后需要根据paper\_id和student\_id进入exam.html页面**



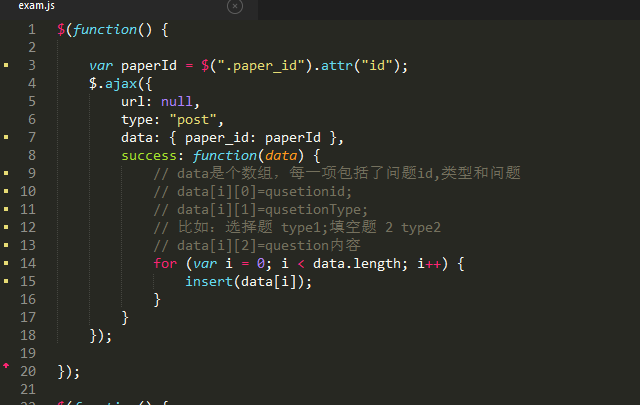
**Exam.js**

根据登录用户初始化界面

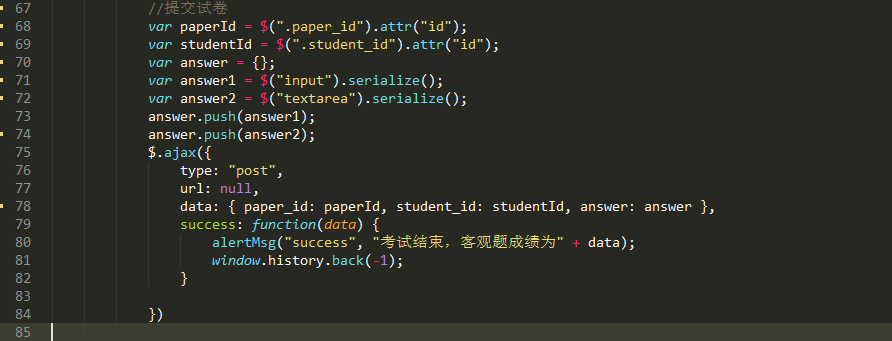


根据paperid请求试卷内容

每一个data[i]是一道小题 包括三个内容



考试结束后，提交信息

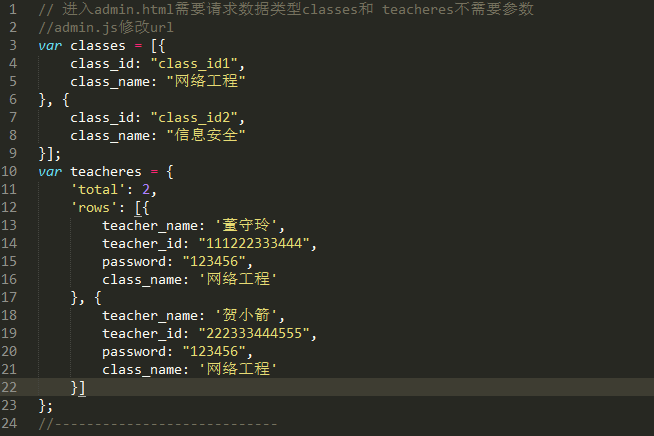


# 管理员部分

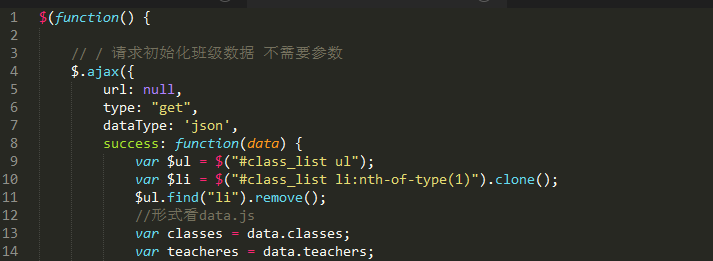
**Admin.js**

**管理员界面初始化数据：**

**Data是个对象data.classes（数组）和data.teachers（包装成下面那样！！！）**



**修改请求url！！！**



**新添班级：浏览器post将新添class\_id和class\_name提交【比如：{"class\_id":"CS-101","class\_name":"计算机科学与技术"}】，服务器检查是否已经存在，若成功；返回msg.success为true,否则为false**



**Student\_dg.js学生信息数据面板**

**根据班级class\_id请求相应班级信息，请返回数据data如下：**

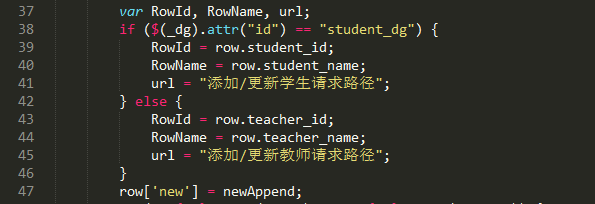


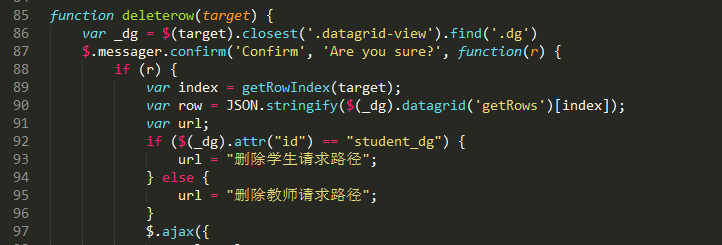
**注意!!形式为对象，total和rows不能少，数据为0时也要!请修改下面请求url**

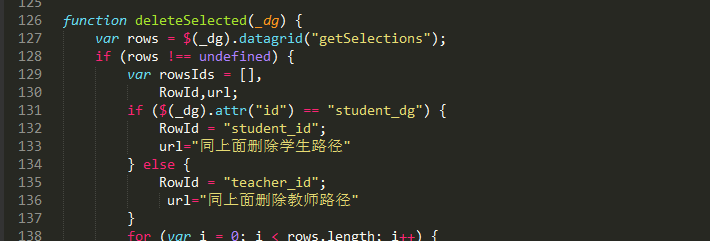


**Datagrid.js**

**1.修改下面请求路径**

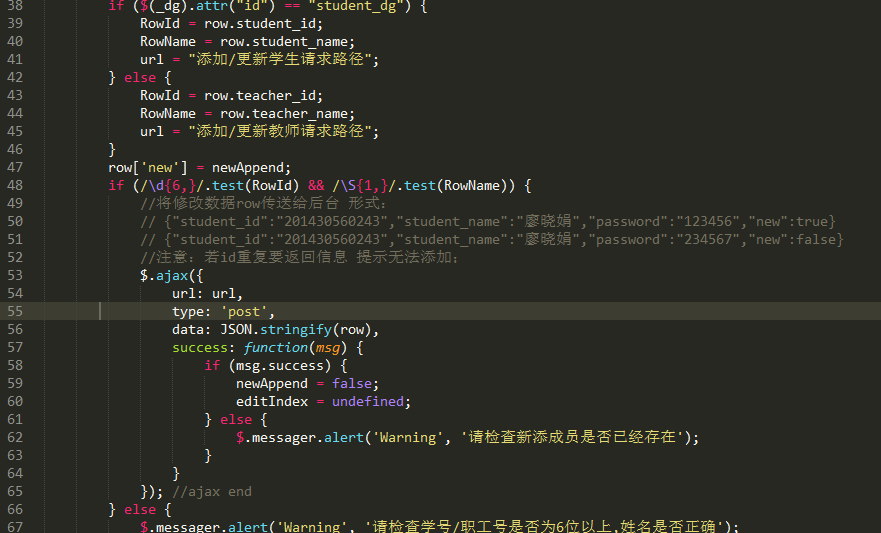






**具体：**

**浏览器提交添加/更新数据：**



**new字段表示添加还是更新，如果是new:true要判断该student\_id是否存在,**

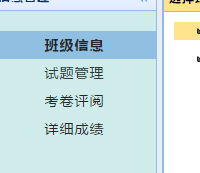
**删除信息：**



**浏览器会提交如上信息：请判断是否存在，返回删除结果，成功的话**

**Msg.success为true**

# 教师部分

这四部分都会要求下面的操作，

【ps：如果可以在浏览器使用session保存登录信息的话，这一部分其实就可以省略了】



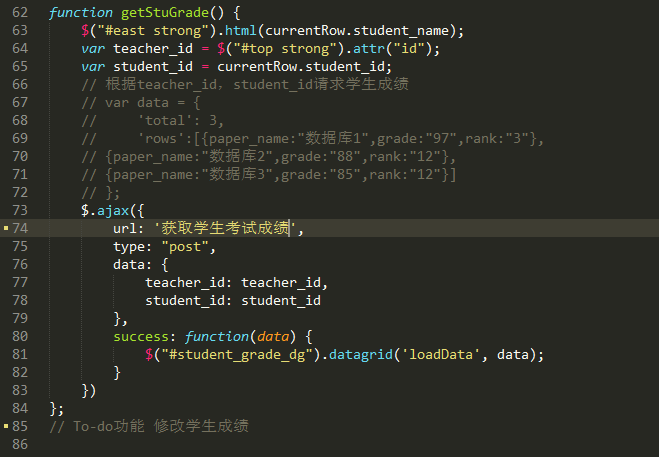
**classManage.js**

根据教师teacher\_id请求该教师所教班级信息

返回数组形式如下：

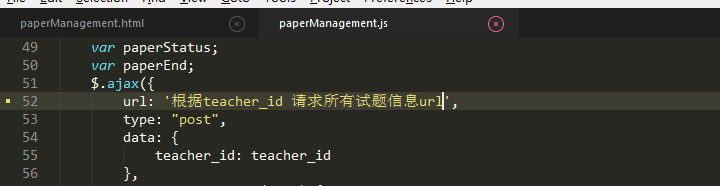
根据class\_id请求班级信息



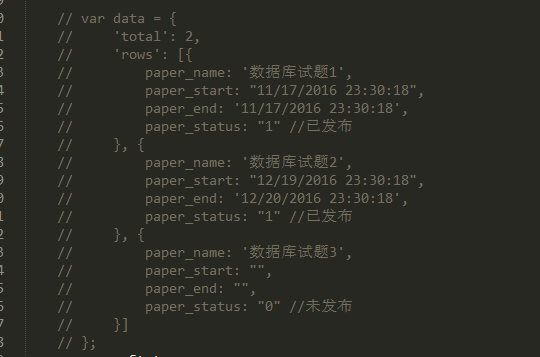


**PaperManagement.html**

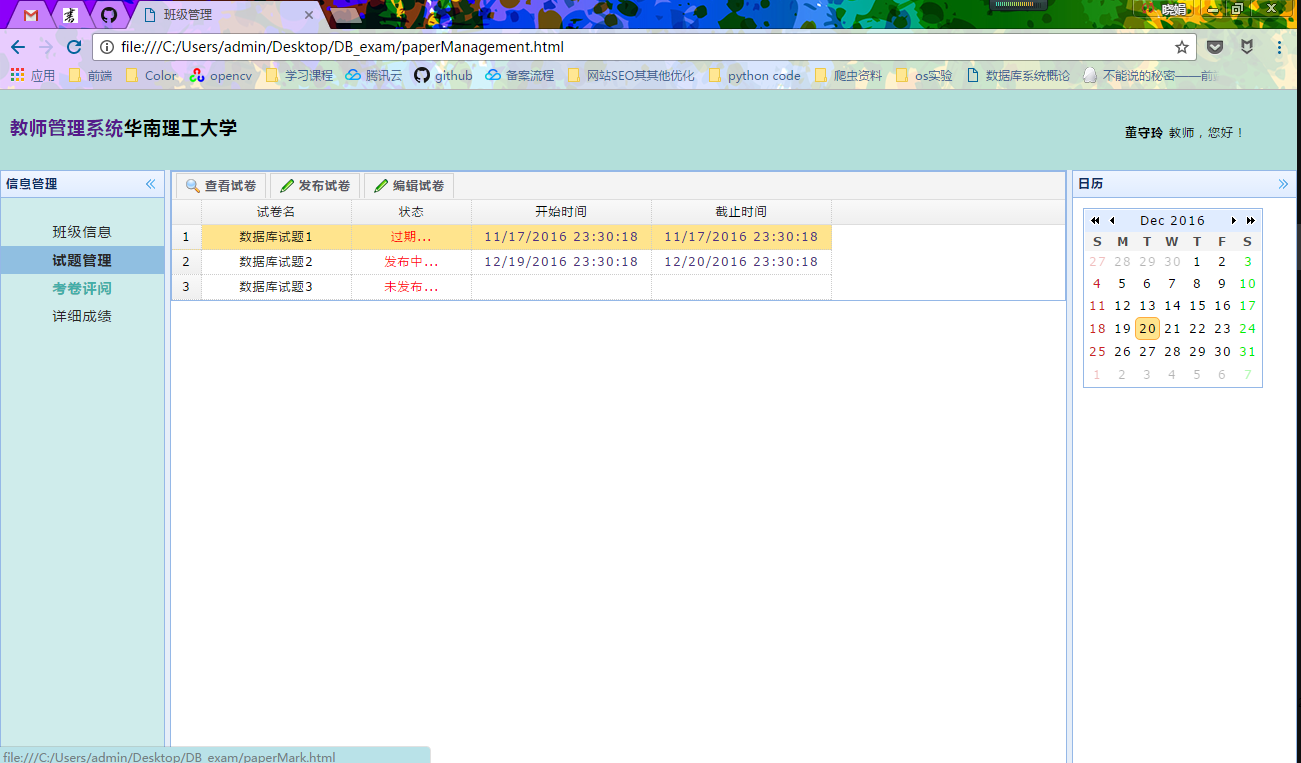
1. 页面加载完成，发起请求：



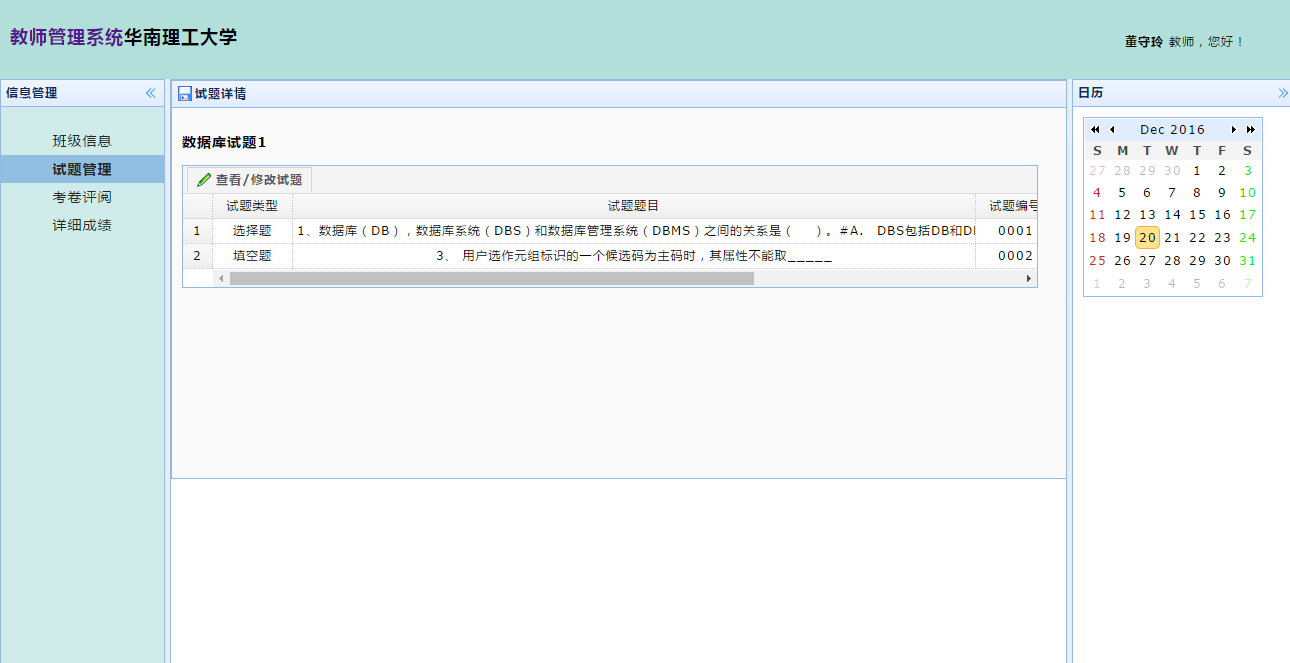
成功返回数据：

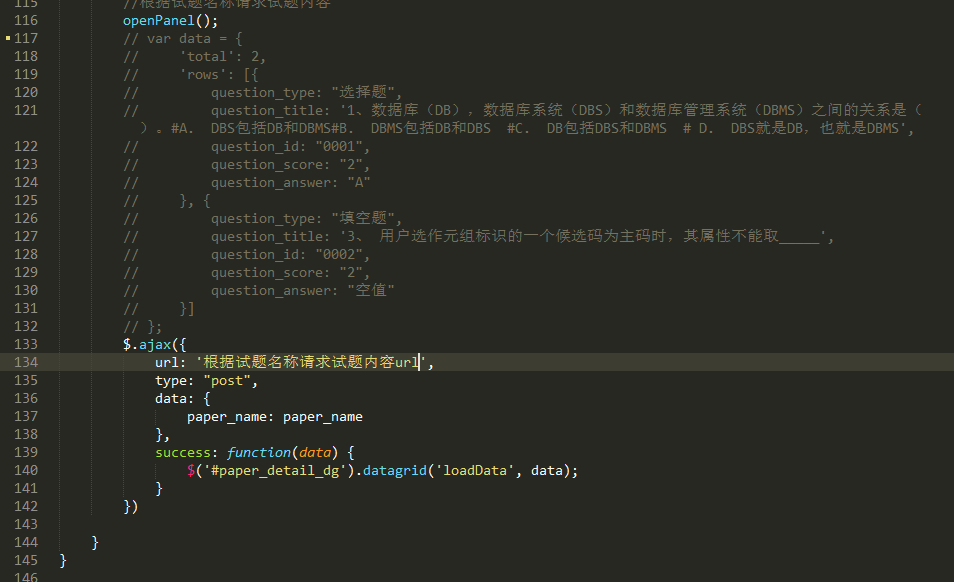


页面初始化后效果如下：

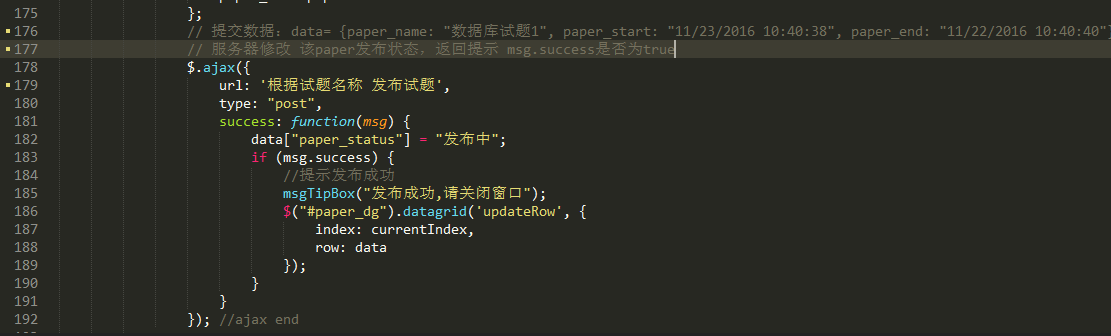


1. 选择试题->点击‘查看试卷’->**发起请求**->进入试卷详情->



**修改url 返回数据如上：**

1. 发布试题

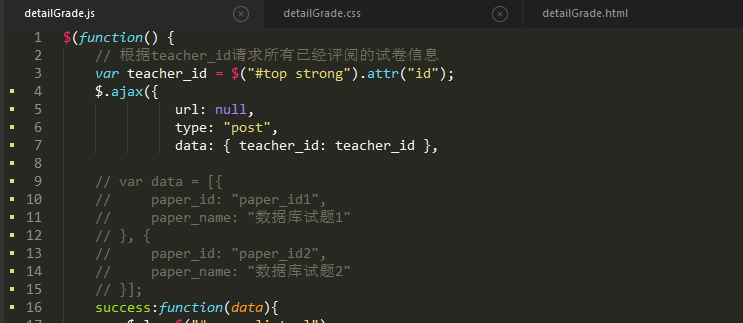


1. 修改试题



**DetailGrade.js**

1. 看图！



1. 看图：



根据考卷获取参与考试的学生成绩



点击成绩，自动排序



最后一部分。。。。。

**paperMark.js**



