文章编号:2095-2163(2019)05-0217-04 中图分类号:TP315 文献标志码:A

基于 jQuery+Ajax 的 ECharts 在成绩统计中的实现

袁江琛
(无锡城市职业技术学院,江苏 无锡 214153)

摘 要: 成绩统计是学校教学工作中一项重要的内容,本文使用 jQuery+Ajax 技术实现成绩查询,并通过 ECharts 图表的形式呈现出来,在查询页面实现异步更新,减少网络数据传输量,提供美观的数据图表,大大增强了用户体验。

关键词: jQuery; Ajax; JSON; ECharts; 成绩统计

Implementation of ECharts in achievement statistics based on jQuery+Ajax

YUAN Jiangchen
(Wuxi City College of Vocational Technology, Wuxi Jiangsu 214153, China)

【Abstract】 Achievement statistics is an important part of school teaching. jQuery + Ajax technology is used to realize achievement query in the paper. The query result is presented in the form of ECharts. Asynchronous updates on query pages can reduce the amount of network data transmission, provide beautiful data charts, and enhance user experience greatly.

【Key words】 jQuery; Ajax; JSON; ECharts; achievement statistics

0 引 言

在成绩管理系统中,可以通过图表的形式直观地显示成绩统计情况,ECharts 是一个用 JavaScript 实现的开源可视化库,提供了折线图、柱状图、散点图、饼图等各种图表,涵盖各行业图表的需求,在成绩管理系统中可以通过 ECharts 个性化定制数据图表,使学生成绩以各种形式进行统计,方便教师和管理员进行分析。

1 相关技术

1.1 jQuery

jQuery 是一个优秀的 JS 代码库,提倡的是"写的少,做的多"的理念,把 JS 中常用功能封装起来,提供了一个快捷的 JS 框架。jQuery 的选择器可以方便地获取页面的各个元素,并提供了各种页面事件,消除了各浏览器之间的兼容性问题,同时 jQuery 对 Ajax 进行了封装,使用户可以更灵活地实现页面异步更新。

1.2 Ajax

Ajax 是一种创建网页动态交互的技术,该技术可以使网页在没有重新加载整个页面的情况下,有选择性地更新其中部分内容,与传统的页面更新不同的是,这种更新只需要页面与服务器之间进行少量的数据交换,实现页面的异步更新,数据传输少,更新效率高,用户体验好。在 jQuery 中,可以使用 $.get(), $.post(), $.ajax() 等方法创建 XMLHttpRequest 对象,对于后台处理的结果可以以 XML、html、JSON、text 等方式返回,前台对于处理结果呈现在页面上,实现页面异步更新。

1.3 JSON 格式

Ajax 处理返回的信息很多时候以 JSON 类型返回。JSON、即 JavaScript 对象表示法,是一种可以通过 JavaScript 解析的结构化数据,任何类型的数据都可以通过 JSON 来表示,其中对象和数组是常用的 2 种类型。对象表示为键值,用大括号来保存对象,用中括号来保存数组,数据之间用逗号分隔。JSON 数据格式语法简单,具有较好的扩展性,结构简单明了,易于操作,在数据传输时比 XML 具有更高的效率,减少网络传输的压力。

2 ECharts 图表

ECharts 来自 Enterprise Charts,是商业级数据图表,具有提供漂亮的图形界面、使用简单、种类繁多、兼容性好等优点,可以流畅地运行在 PC 端和移动设备上,同时 ECharts 可以实现异步数据更新,在初始化图表后可以通过 jQuery 工具异步获取数据再通过 setOption 填入数据和配置相关项即可,其中 setOption 使用 JSON 数据格式。

作者简介:袁江琛(1974-),女,硕士,副教授,主要研究方向:软件技术。

收稿日期:2019-07-14

3 系统实现

本文以初一年级8个班的4门课程为例,可以统计某个班级单科成绩、全年级某门课程和某个班级所有课程汇总信息,下面拟重点阐释论述某班单科成绩统计的设计实现。

3.1 数据库设计

成绩统计是教学管理系统中的一部分,本文只讨论成绩统计这一模块,列举该模块需要用到的数据表。该模块研发中将涉及到的数据表有:学生表、班级表、课程表和成绩表。其中,学生表中主要字段有学生学号、姓名、所在班级编号等;班级表中主要字段有班级编号、班级名称等;课程表中主要字段有课程编号、课程名称等;成绩表中主要字段有成绩编号、学生学号、课程编号、成绩等。

3.2 某班单科成绩统计

当要选择查看某个班级某门课程的统计情况,可以选择班级和课程,前台将班级编号和课程编号传送给后台,后台根据这2个条件对数据库执行查询操作,将查询结果以JSON数据返回前台,前台接收数据后将该班学生某门课程的成绩以表格的形式表示,同时将各分数段的数据填入相应单元格中,并将这些统计数据以饼图的方式加以呈现。这里,关于该部分的设计研究内容详见如下。

(1)后台程序设计。后台相应的代码如下:

```
public void getScore(HttpContext context)
{
    string s="";
    string sqlStr = " select ClassName, StudentName, SubjectName, ScroeMark from v_Score where classid = @ ClassID and subjectid = @ SubjectID";
    SqlParameter[] para = { new SqlParameter("@ClassID", context.Request.Form["ClassID"]), new SqlParameter("@SubjectID", context.Request.Form["SubjectID"]) };
    SqlDataReader sda = DB.ExecuteReader(sqlStr, CommandType.Text, para);
    string item="";
    while(sda.Read())
    {
        if(item != "")item += ",";
        item += "{\"ClassName\":\"" + sda[0].ToString() + "\",\"StudentName\":\"" + sda[1].ToString() + "\",\"SubjectName\":\"" + sda[2].ToString() + "\",\"ScroeMark\":\"" + sda[3].ToString() + "\"}";
    }
    s += "{\"Items\":[" + item + "],";
    sqlStr= " select count ( case when ScroeMark between 90 and 100 then 1 end) as[90-100分], count(case when ScroeMark between 80 and 89 then 1 end) as[80-89分], count(case when ScroeMark between 70 and 79 then 1 end) as[70-79分],count(case when ScroeMark between 60 and 69 then 1 end) as[60-69分], count(case when ScroeMark<60 then 1 end) as[不及格] from v_Score where classid = @ ClassID and subjectid = @ SubjectID";
    sda = DB.ExecuteReader(sqlStr, CommandType.Text, para);
    sda.Read();
    s += "\"S9\":\""+sda[0].ToString()+"\",\"S8\":\""+sda[1].ToString()+"\",\"S7\":\""+sda[2].ToString()+"\",\"S6\":\""+sda[3].ToString()+"\",\"S5\":\""+sda[4].ToString()+"\"}";
    context.Response.Write(s);
}
```

(2)前台软件设计。前台处理后台数据主要分3步,研究可得解析分述如下。

① 将学生的成绩填入表格中,相关代码如下:

```
$("#ScroeMark table").append("<tr><th>班级</th><th>姓名</th><th>课程</th><th>分数</th><th>班级</th><th>姓名</th><th>课程</th><th>分数</th></tr>");
for (var i=0; i< 25;i++) {
    var ss="";
    if (i>=a.Items.length) {
        var ss="<tr><td></td><td></td><td></td><td></td><td></td>" +
        "<td></td><td></td><td></td><td></td><td></td></tr>";
    }
    else if ((i + 25) < a.Items.length) {
        var ss = "<tr><td>" + a.Items[i].ClassName + "</td><td>" + a.Items[i].SubjectName + "</td><td>" + a.Items[i].
```

```
StudentName + " </td > < td >" + a. Items [ i ]. ScroeMark + "</td>" +
    "<td>" + a. Items[ i+25 ]. ClassName + "</td><td>" + a. Items[ i+25 ]. SubjectName+"</td><td>" + a. Items[ i + 25 ]. StudentName + "</td><td>" + a. Items[ i + 25 ]. ScroeMark + "</td></tr>";
    }
    else {
      var ss = "<tr><td>" + a. Items [ i ]. ClassName + "</td><td>" + a. Items [ i ]. SubjectName + "</td><td>" + a. Items [ i ]. StudentName + "</td><td>" + a. Items [ i ]. ScroeMark + "</td>" +"<td></td><td></td><td></td><td></td><td></td></tr>";
    }
    $("#ScroeMark table"). append(ss);
  }
```

② 将统计结果填入相应单元格，代码略。

③ 生成图表，在该统计模块中使用饼图呈现统计结果。首先对图表的相关数据进行配置，如图表的标题，数据值的表示方式、数据段的分步以及各个数据分段系列的值，最终使用指定的配置项和数据显示图表。其对应代码如下：

```
if (myChart != null && myChart != "" && myChart != undefined) {
    myChart. dispose();
}
myChart = echarts. init (document. getElementById("eChart"));
//指定图表的配置项和数据
option = {
  title: {
  text: $("#Class"). find("option:selected"). text() + $("#Subject"). find("option:selected"). text(), subtext: '成绩分布', x: 'center'},
    tooltip: {
      trigger: 'item', formatter: "{a} <br/> {b} : {c} ({d}%)"},
    legend: {
      orient: 'vertical', left: 'right', data: ['90-100', '80-89', '70-79', '60-69', '<60']},
    series: [
      {
        name: '成绩', type: 'pie', radius: '80%', center: ['50%', '60%'],
        label: {
          normal: {
            show: true, position: 'inner', formatter: '{b} :\n {c}人\n ({d}%)'}},
        data: [
          {value: s1, name: '90-100'},
          {value: s2, name: '80-89'},
          {value: s3, name: '70-79'},
          {value: s4, name: '60-69'},
          {value: s5, name: '<60'}],
        itemStyle: {
          emphasis: {
            shadowBlur: 10, shadowOffsetX: 0, shadowColor: 'rgba(0, 0, 0, 0.5)'}
        }
      }
    ];
  //使用刚指定的配置项和数据显示图表。
  myChart. setOption(option);
```

运行效果如图 1 所示。由图 1 可知，在对各班级某一门课程的成绩进行统计后，就可以看到每个班级的均分、最高分、最低分和各分数段的人数，统计结果以折线图的方式呈现，运行效果如图 2 所示。

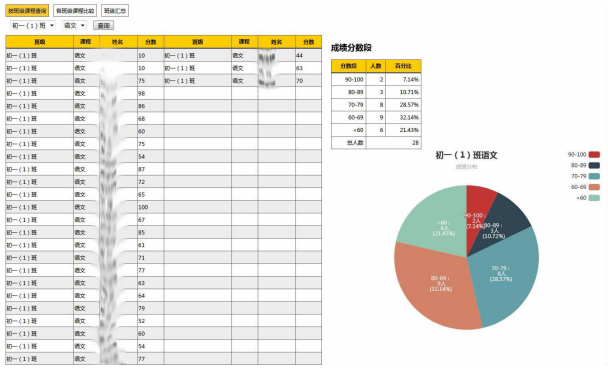

图 1　某班单科成绩统计

Fig. 1　Achievement statistics of a class's single subject

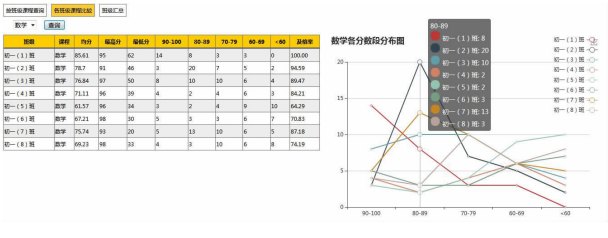

图 2　各班级某课程成绩统计

Fig. 2　Achievement statistics of a subject in each class

（下转第 222 页）