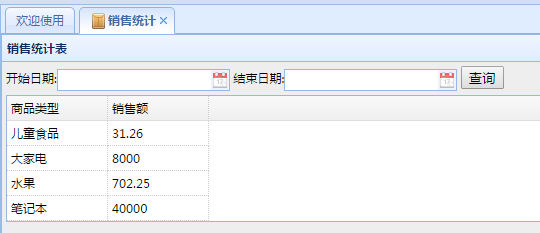
# ERP项目笔记 Day-08

## 报表与图表

1. 销售统计（报表）
   1. 需求及实现思路
      1. 需求

统计各类别商品的销售额，如下图



* + 1. 实现思路

技术：运用关联查询、聚合统计。

关键是SQL编写。

找出涉及的表有哪些？

* 报表数据存在哪个表里？
* 数据存在的表之间的中间表
* 查询条件所在的表
* 关联查询条件所在的表 和 数据所在的表之间 的中间表

销售额来源：订单明细表

商品类别： 商品类别表

关联用的表：商品表

条件类型为销售订单： 订单表

* + 1. 查询语句编写, 由简单到复杂

1. 先将这4个表关联起来查询



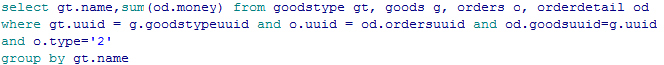


1. 增加订单类型为2的条件，即销售订单



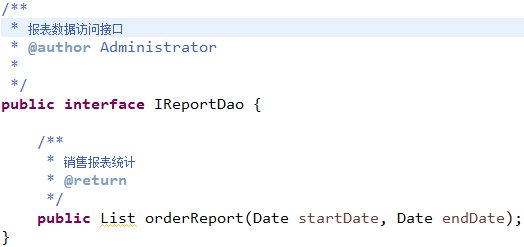


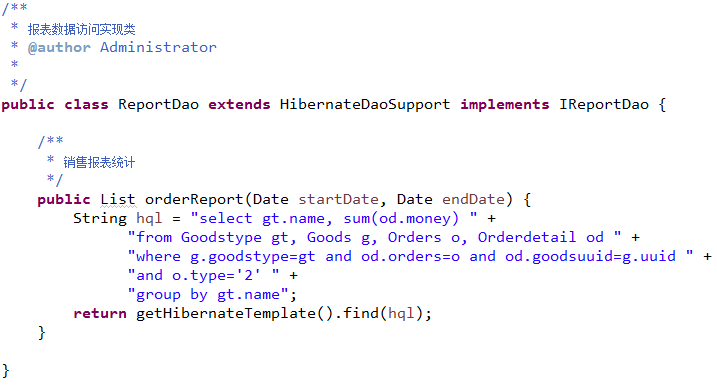
1. 按商品类型名称分组，并统计销售总额





* 1. 后端代码实现
     1. 创建 IReportDao和ReportDao



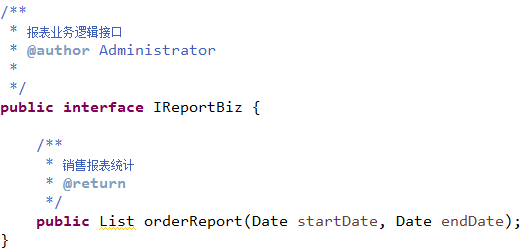


注意关联部分和原生 SQL 不同之处。

* from 后跟的是类名 （大写开头）
* 如果两个实体进行了一对多关联，关联的写法与 sql 不同

多的一方.一的一方属性 = 一的一方的别名

* 特别注意： 当我们的语句比较长，每换行后需要加空格
  + 1. 创建 IReportBiz 和 ReportBiz



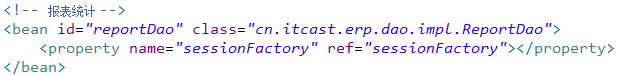


* + 1. 创建 ReportAction

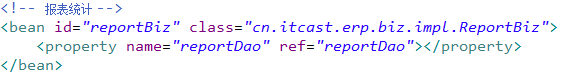
|  |
| --- |
| **package** cn.itcast.erp.action;  **import** java.io.IOException;  **import** java.util.Date;  **import** java.util.List;  **import** javax.servlet.http.HttpServletResponse;  **import** org.apache.struts2.ServletActionContext;  **import** com.alibaba.fastjson.JSON;  **import** cn.itcast.erp.biz.IReportBiz;  /\*\*  \* 统计报表  \* **@author** Administrator  \*  \*/  **public** **class** ReportAction {  **private** Date startDate;  **private** Date endDate;  **private** IReportBiz reportBiz;  /\*\*  \* 销售统计报表  \*/  **public** **void** orderReport(){  List reportData = reportBiz.orderReport(startDate, endDate);  write(JSON.*toJSONString*(reportData));  }  **public** Date getStartDate() {  **return** startDate;  }  **public** **void** setStartDate(Date startDate) {  **this**.startDate = startDate;  }  **public** Date getEndDate() {  **return** endDate;  }  **public** **void** setEndDate(Date endDate) {  **this**.endDate = endDate;  }  **public** **void** setReportBiz(IReportBiz reportBiz) {  **this**.reportBiz = reportBiz;  }    /\*\*  \* 输出给前端  \* **@param** mapString  \*/  **public** **void** write(String mapString) {  //返回对象  HttpServletResponse res = ServletActionContext.*getResponse*();  res.setContentType("text/html;charset=utf-8");  res.setCharacterEncoding("utf-8");  **try** {  //输出给页面  res.getWriter().write(mapString);  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

* + 1. 编写配置文件

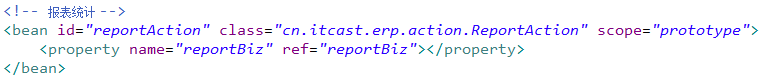
1. 在applicationContext-dao.xml中添加



1. 在applicationContext-service.xml中添加



1. 在applicationContext-action.xml中添加

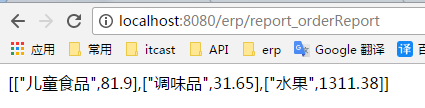


1. 在struts.xml中添加



* + 1. 测试结果

浏览器访问：



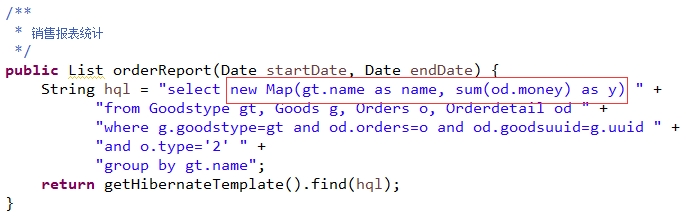
* + 1. 结果分析

这种数据很奇怪，像是一种复合类型的数组，这种数据easyui的datagrid是无法加载的，我们看回datagrid加载的数据格式都是json数据{key:value}。因此我们要想方法把它转换成json数据格式

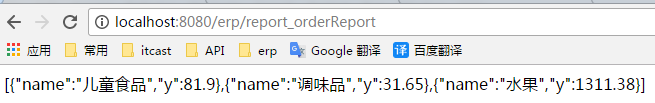
* + 1. 将查询结果封装成map数据

我们可以使用投影查询来封装我们的查询结果：

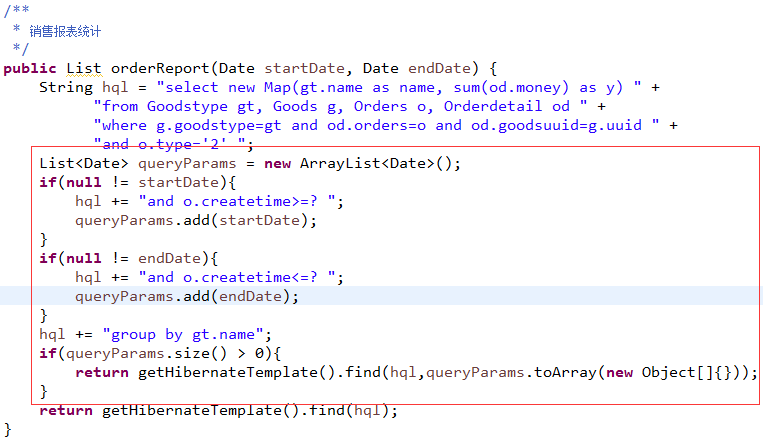
修改 ReportDao 类中 OrderReport 方法



运行后结果如下：

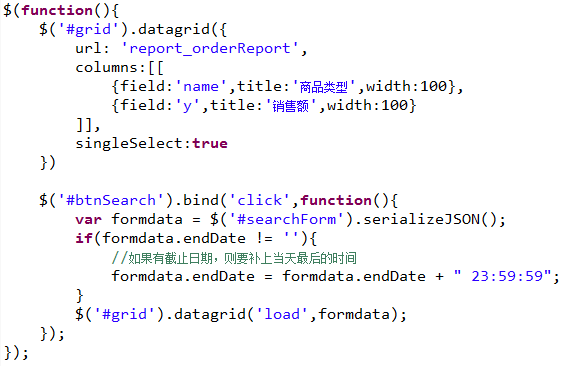


* + 1. 修改ReportDao的OrderReport方法，支持日期区间查询



* 1. 前端实现

1. 在js目录下创建report.js



1. 创建report\_order.html



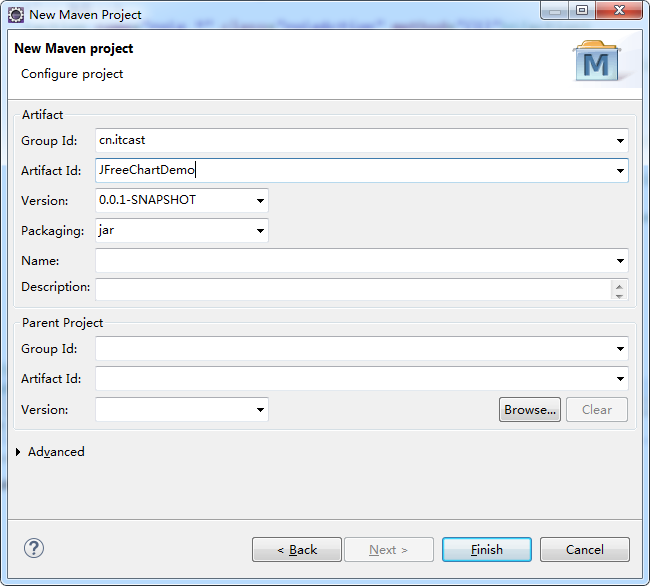
1. 图表工具入门
   1. JFreeChart

JFreeChart是JAVA平台上的一个开放的图表绘制类库。它完全使用JAVA语言编写，是为applications, applets, servlets 以及JSP等使用所设计。JFreeChart可生成饼图（pie charts）、柱状图（bar charts）、[散点图](http://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE" \t "http://baike.baidu.com/_blank)（scatter plots）、[时序图](http://baike.baidu.com/item/%E6%97%B6%E5%BA%8F%E5%9B%BE" \t "http://baike.baidu.com/_blank)（time series）、[甘特图](http://baike.baidu.com/item/%E7%94%98%E7%89%B9%E5%9B%BE" \t "http://baike.baidu.com/_blank)（Gantt charts）等等多种图表，并且可以产生PNG和[JPEG格式](http://baike.baidu.com/item/JPEG%E6%A0%BC%E5%BC%8F" \t "http://baike.baidu.com/_blank)的输出，还可以与PDF和EXCEL关联。

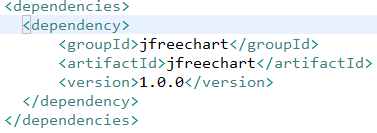
1. 入门demo

在 D 盘根目录生成饼图图片 chart.png

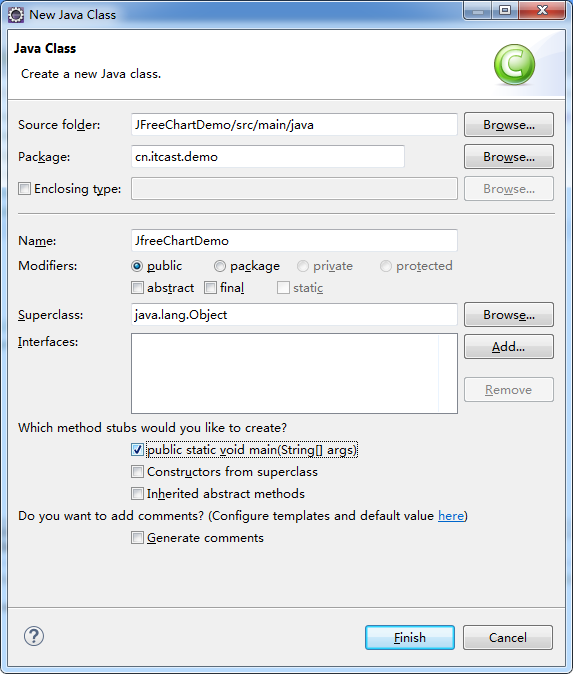
1. 创建JfreeChartDemo工程



1. 引入坐标



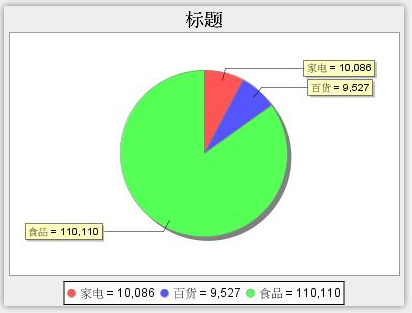
1. 创建demo类



1. 编写代码

|  |
| --- |
| **package** cn.itcast.demo;  **import** java.io.File;  **import** org.jfree.chart.ChartFactory;  **import** org.jfree.chart.ChartUtilities;  **import** org.jfree.chart.JFreeChart;  **import** org.jfree.data.general.DefaultPieDataset;  **public** **class** JfreeChartDemo {  **public** **static** **void** main(String[] args) **throws** Exception {  DefaultPieDataset dataset = **new** DefaultPieDataset();  dataset.setValue("家电", 10086);  dataset.setValue("百货", 9527);  dataset.setValue("食品", 110110);  // 参数 1 title 标题  // 参数 2 dataset 数据集  // 参数 3 是否开启图例  // 参数 4 是否开启工具栏  // 参数 5 是否开启 url 跳转  // 创建一张饼图  JFreeChart chart = ChartFactory.*createPieChart*("标题", dataset, **true**, **false**, **false**);  // 参数 1 生成图片文件到本地  // 参数 2 chart对象  // 参数 3 图片宽度  // 参数 4 图片高度  ChartUtilities.*saveChartAsJPEG*(**new** File("d:\\chartDemo.jpeg"), chart, 400, 300);  }  } |

1. 查看结果



* 1. HighCharts

Highcharts 是一个用纯[JavaScript](http://baike.baidu.com/item/JavaScript" \t "http://baike.baidu.com/item/_blank)编写的一个图表库， 能够很简单便捷的在web网站或是web应用程序添加有交互性的图表，并且免费提供给个人学习、个人网站和非商业用途使用。HighCharts支持的图表类型有曲线图、区域图、柱状图、饼状图、散状点图和综合图表。

HighCharts  界面美观，由于使用JavaScript编写，所以不需要像Flash和Java那样需要插件才可以运行，而且运行速度快。另外HighCharts还有很好的兼容性，能够完美支持当前大多数浏览器

我们的ERP项目就采用HighCharts

1. 入门Demo

看项目资料里的示例

1. 销售统计图（饼图）
   1. 需求

将销售统计表的结果，以饼图的形式展现



* 1. 代码实现

1. 将highchars.js 拷贝到ui文件夹，并在页面 report\_order.html 中引入highchars.js



1. 修改页面report\_order.html

|  |
| --- |
| <body class=*"easyui-layout"*>  <div data-options=*"region:'center',title:'销售统计表'"* style="padding:*4px*;background-color:*#eee*">  <form id=*"searchForm"*>  开始日期:<input name=*"startDate"* class=*"easyui-datebox"*>  结束日期:<input name=*"endDate"* class=*"easyui-datebox"*>  <button type=*"button"* id=*"btnSearch"*>查询</button>  </form>  <div style="height:*4px*;"></div>  <table id=*"grid"*></table>  </div>  <div data-options=*"region:'east',title:'销售统计图',split:true"* style="width:6*00px*;">  <div id=*"pieChart"*></div>  </div>  </body> |

1. 修改 report.js,添加 showChart()函数



属性解释：

* chart ：图表基本属性

a) plotBackgroundColor 区域背景颜色

b) plotBorderWidth 区域边框宽度

c) plotShadow 区域阴影

d) type 图表类型

* title ：图表标题

a) text 标题文本

* tooltip ：工具提示

a) pointFormat 工具提示显示格式

* plotOptions：区域选项

a) allowPointSelect 点击区域后选择

b) cursor 光标类型

c) dataLabels 数据标签

d) showInLegend 是否显示图例

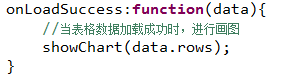
* series： 数据组

a) name 名称

b) colorByPoint 点的颜色

c) data 数据

1. 修改report.js，给grid表格添加onLoadSuccess事件

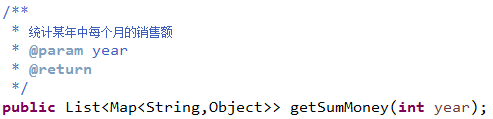


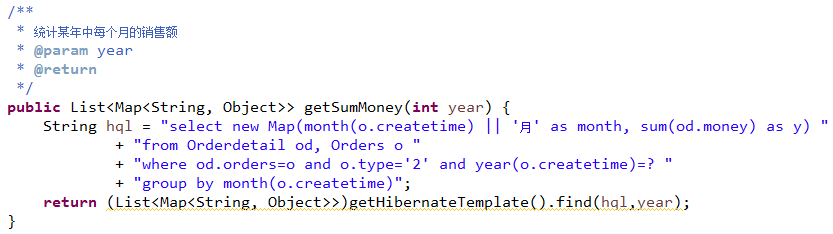
1. 销售趋势分析（报表）
   1. 需求

选择年份点击查询，可以看到所选择年份的各个月份的销售额的统计数据

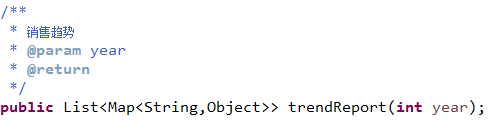


* 1. 后端实现
     1. 给IReportDao和ReportDao添加方法



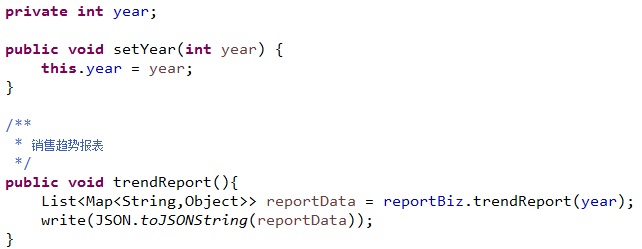


* + 1. 给IReportBiz和ReportBiz添加方法

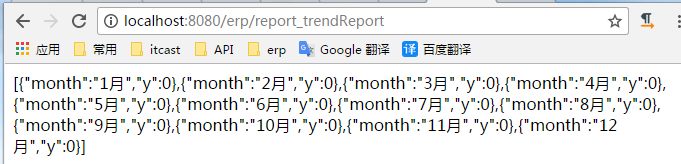


|  |
| --- |
| /\*\*  \* 销售趋势  \* **@param** year  \* **@return**  \*/  **public** List<Map<String,Object>> trendReport(**int** year){  //保存每个月份的销售额  List<Map<String, Object>> result = **new** ArrayList<Map<String,Object>>(12);  //获取当年每月销售额  List<Map<String,Object>> yearData = reportDao.getSumMoney(year);  //把从DB中得到的当年每月销售额转换成map集合，用于下面的查缺补漏，因为可能存在某些月份没有销售额的情况  Map<String, Map<String,Object>> map = **new** HashMap<String,Map<String,Object>>();  **for**(Map<String,Object> m : yearData){  map.put((String)m.get("month"), m);  }  Map<String,Object> data = **null**;  //按12个月，对每个月份的数据进行封装，最终以List<Map<String,Object>>形式返回  **for**(**int** i = 1; i <= 12; i++){  data = map.get(i + "月");  **if**(**null** == data){  //如果当月没有销售额，则补上当月的月份和数据0  data = **new** HashMap<String,Object>();  data.put("month", i+"月");  data.put("y", 0);  }  result.add(data);  }  **return** result;  } |

* + 1. 编写Action代码



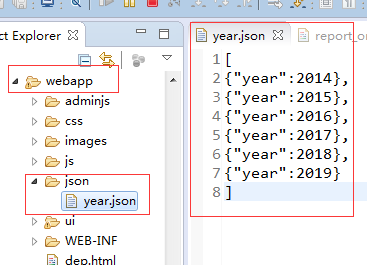
* + 1. 浏览器测试结果



* 1. 前端实现
     1. 添加report\_trend.js

|  |
| --- |
| $(**function**(){  $('#grid').datagrid({  queryParams:{},  columns:[[  {field:'month',title:'月份',width:100},  {field:'y',title:'销售额',width:100}  ]],  singleSelect:**true**  });  $('#btnSearch').bind('click',**function**(){  **var** submitData = $('#searchForm').serializeJSON();  //提交查询条件，远程重新加载数据，  $('#grid').datagrid({  url:'report\_trendReport',  queryParams:submitData  });  });  }); |

* + 1. 在webapp下创建json目录，并创建year.json文件

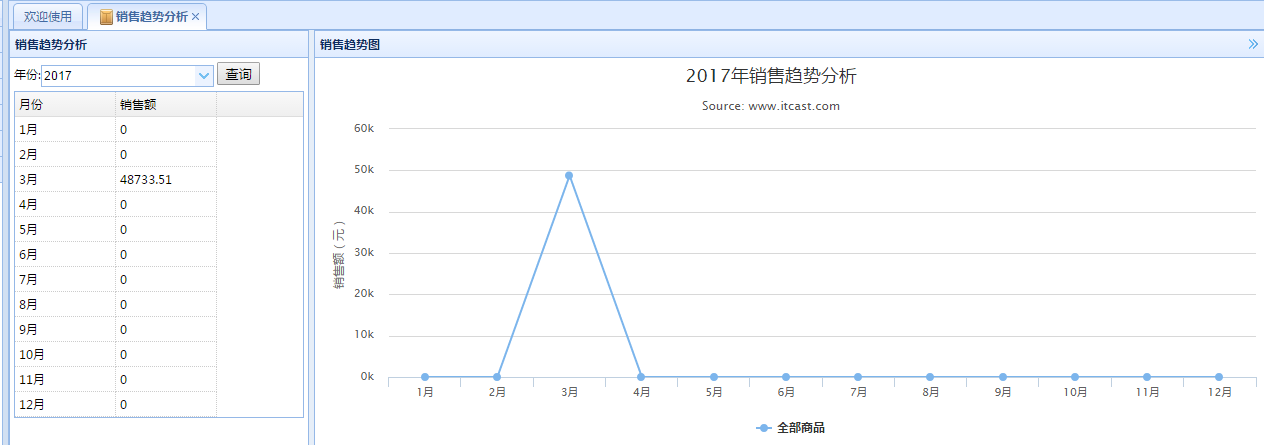


* + 1. 添加report\_trend.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>销售趋势报表</title>  <link rel=*"stylesheet"* type=*"text/css"* href=*"ui/themes/default/easyui.css"*>  <link rel=*"stylesheet"* type=*"text/css"* href=*"ui/themes/icon.css"*>  <script type=*"text/javascript"* src=*"ui/jquery.min.js"*></script>  <script type=*"text/javascript"* src=*"ui/jquery.easyui.min.js"*></script>  <script type=*"text/javascript"* src=*"ui/locale/easyui-lang-zh\_CN.js"*></script>  <script type=*"text/javascript"* src=*"ui/jquery.serializejson.min.js"*></script>  <script type=*"text/javascript"* src=*"ui/highcharts.js"*></script>  <script type=*"text/javascript"* src=*"js/report\_trend.js"*></script>  </head>  <body>  <div style="padding-left:*4px*;">  <div style="height:*2px*;"></div>  <form id=*"searchForm"*>  年份:<input name=*"year"* id=*"year"* class=*"easyui-combobox"* data-options=*"*  *url:'json/year.json',valueField:'year',textField:'year'*  *"*>  <button type=*"button"* id=*"btnSearch"*>查询</button>  </form>  <div style="height:*2px*;"></div>  <table id=*"grid"*></table>  </div>  </body>  </html> |

1. 销售趋势分析（折线图）
   1. 需求

在上边销售趋势分析数据的基础上，显示折线图



* 1. 代码实现
     1. 修改report\_trend.html

|  |
| --- |
| <body class=*"easyui-layout"*>  <div data-options=*"region:'center',title:'销售趋势报表'"* style="padding:*4px*;background-color:*#eee*">  <div style="height:*2px*;"></div>  <form id=*"searchForm"*>  年份:<input name=*"year"* id=*"year"* class=*"easyui-combobox"* data-options=*"*  *url:'json/year.json',valueField:'year',textField:'year'*  *"*>  <button type=*"button"* id=*"btnSearch"*>查询</button>  </form>  <div style="height:*2px*;"></div>  <table id=*"grid"*></table>  </div>  <div data-options=*"region:'east',title:'销售趋势图',split:true"* style="width:*700px*;">  <div id=*"trendChart"*></div>  </div>  </body> |

* + 1. 修改report\_trend.js

1. 添加showChart方法

|  |
| --- |
| /\*\*  \* 画销售趋势图  \*/  **function** showChart(){  **var** monthData = **new** Array();  **for**(**var** i = 1; i <=12; i++){  monthData.push(i+"月");  }  $('#trendChart').highcharts({  title: {  text: $('#year').combobox('getValue') + '年销售趋势分析',  x: -20 //center  },  subtitle: {  text: 'Source: www.itcast.com',  x: -20  },  xAxis: {  categories: monthData  },  yAxis: {  title: {  text: '销售额（元）'  },  plotLines: [{  value: 0,  width: 1,  color: '#808080'  }]  },  tooltip: {  valueSuffix: '元'  },  legend: {  layout: 'vertical',  align: 'center',  verticalAlign: 'bottom',  borderWidth: 0  },  series: [{name:'全部商品',data:$('#grid').datagrid('getRows')}]  });  } |

1. 当grid加载成功后调用showChart方法展示趋势图

