# 市级医药采购系统统计分析

## 功能模块--区域统计:

### 需求分析:

用户通过按区域统功能实现按区域统计每个区域的下卫生室的金额总和,通过导出Excel功能将统计结果导出成Els



#### 相关技术:

Echarts前端框架

### 功能实现:

#### Jsp页面:

|  |
| --- |
| <!--Step:1 Prepare a dom for ECharts which (must) has size (width & hight)-->  <!--Step:1 为ECharts准备一个具备大小（宽高）的Dom-->    <div class=*"navbar navbar-default navbar-fixed-top"* role=*"navigation"* id=*"head"*></div>    <div id=*"main"* style="height:*500px*;border:*1px solid #ccc*;padding:*10px*;">    </div>  <!--Step:2 Import echarts.js-->  <!--Step:2 引入echarts.js-->  <script src=*"js/echarts.js"*></script>  <!--加载主题-->  <script src=*"js/zhuti/macarons.js"*></script>  <script type=*"text/javascript"*>      $(**function**(){  $('#startDate').datebox({  onSelect: **function**(date){  **var** startDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#startDate').val(startDate);  }  });  $('#endDate').datebox({  onSelect: **function**(date){  **var** endDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#endDate').val(endDate);  }  });  })    **function** query(){      // Step:3 conifg ECharts's path, link to echarts.js from current page.  // Step:3 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需图表路径  require.config({  paths: {  echarts: './js'  }  });  // Step:4 require echarts and use it in the callback.  // Step:4 动态加载echarts然后在回调函数中开始使用，注意保持按需加载结构定义图表路径  require(  [  'echarts',  'echarts/chart/bar',  'echarts/chart/line'  ],  **function**(ec) {  //--- 声明一个折柱 ---  myChart = ec.init(document.getElementById('main'), 'macarons');  initData();//加载数据  // --- 地图 ---  /\* 读取数据时加载效果 \*/  //myChart.showLoading();  //initData();//加载数据    }  );  }    query();    /\* 加载数据ajax \*/  **function** initData(){    $.ajax({  type : "post",  async : **true**, //异步请求（同步请求将会锁住浏览器，用户其他操作必须等待请求完成才可以执行）  url : "${baseurl}/analyze/AjaxSumbyarea.action", //请求发送到TestServlet处  data : $('#businesslistForm').serialize(),  dataType : "json", //返回数据形式为json  success : **function**(result) {  //请求成功时执行该函数内容，result即为服务器返回的json对象  //清空画布，防止缓存---------------------------------  myChart.clear();    myChart.setOption(result);    myChart.hideLoading();  //如果不清空,图形数据将不会更改        },  error : **function**(errorMsg) {  //请求失败时执行该函数  alert("图表请求数据失败!");  }  })  }    </script> |

#### Action实现:

|  |
| --- |
| <!--Step:1 Prepare a dom for ECharts which (must) has size (width & hight)-->  <!--Step:1 为ECharts准备一个具备大小（宽高）的Dom-->    <div class=*"navbar navbar-default navbar-fixed-top"* role=*"navigation"* id=*"head"*></div>    <div id=*"main"* style="height:*500px*;border:*1px solid #ccc*;padding:*10px*;">    </div>  <!--Step:2 Import echarts.js-->  <!--Step:2 引入echarts.js-->  <script src=*"js/echarts.js"*></script>  <!--加载主题-->  <script src=*"js/zhuti/macarons.js"*></script>  <script type=*"text/javascript"*>      $(**function**(){  $('#startDate').datebox({  onSelect: **function**(date){  **var** startDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#startDate').val(startDate);  }  });  $('#endDate').datebox({  onSelect: **function**(date){  **var** endDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#endDate').val(endDate);  }  });  })    **function** query(){      // Step:3 conifg ECharts's path, link to echarts.js from current page.  // Step:3 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需图表路径  require.config({  paths: {  echarts: './js'  }  });  // Step:4 require echarts and use it in the callback.  // Step:4 动态加载echarts然后在回调函数中开始使用，注意保持按需加载结构定义图表路径  require(  [  'echarts',  'echarts/chart/bar',  'echarts/chart/line'  ],  **function**(ec) {  //--- 声明一个折柱 ---  myChart = ec.init(document.getElementById('main'), 'macarons');  initData();//加载数据  // --- 地图 ---  /\* 读取数据时加载效果 \*/  //myChart.showLoading();  //initData();//加载数据    }  );  }    query();    /\* 加载数据ajax \*/  **function** initData(){    $.ajax({  type : "post",  async : **true**, //异步请求（同步请求将会锁住浏览器，用户其他操作必须等待请求完成才可以执行）  url : "${baseurl}/analyze/AjaxSumbyarea.action", //请求发送到TestServlet处  data : $('#businesslistForm').serialize(),  dataType : "json", //返回数据形式为json  success : **function**(result) {  //请求成功时执行该函数内容，result即为服务器返回的json对象  //清空画布，防止缓存---------------------------------  myChart.clear();    myChart.setOption(result);    myChart.hideLoading();  //如果不清空,图形数据将不会更改        },  error : **function**(errorMsg) {  //请求失败时执行该函数  alert("图表请求数据失败!");  }  })  }    </script> |

#### 效果图:

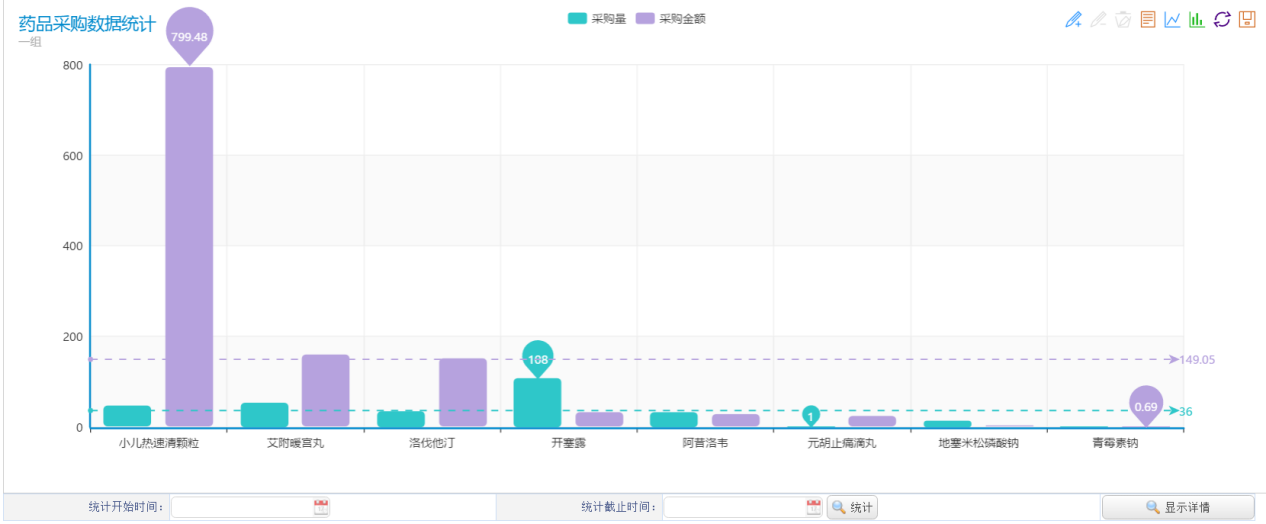


需求2:



实现2:





## 功能模块--药品统计:

### 需求分析:

按药品统计采购量以及采购金额,卫生局和卫生院只能查看自己管辖地区的采购单统计

要求可以根据日期查询相应时间段信息

#### 相关技术:

Echarts前端框架

### 代码实现:

#### Jsp页面:

|  |
| --- |
| <!-- --------------------------- -->  <script type=*"text/javascript"*>  **function** query(){    // Step:3 conifg ECharts's path, link to echarts.js from current page.  // Step:3 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需图表路径  require.config({  paths: {  echarts: '/business/analyze/js'  }  });  // Step:4 require echarts and use it in the callback.  // Step:4 动态加载echarts然后在回调函数中开始使用，注意保持按需加载结构定义图表路径  require(  [  'echarts',  'echarts/chart/bar',  'echarts/chart/line'  ],  **function**(ec) {  //--- 声明一个折柱 ---  myChart = ec.init(document.getElementById('main'), 'macarons');  initData();//加载数据  // --- 地图 ---  /\* 读取数据时加载效果 \*/  myChart.showLoading();  initData();//加载数据    }  );  }    /\* 加载数据ajax \*/  **function** initData(){    //表单参数序列化  **var** params = $("#drugyzreaqueryForm").serializeJson();  //刷新datagride  $("#drugyzrealist").datagrid('reload',params);    $.ajax({  type : "post",  async : **false**, //异步请求（同步请求将会锁住浏览器，用户其他操作必须等待请求完成才可以执行）  url : "${baseurl}/analyze/ajaxDrugyzrea\_bar.action", //请求发送到TestServlet处  data : params,  dataType : "json", //返回数据形式为json  success : **function**(result) {  //请求成功时执行该函数内容，result即为服务器返回的json对象  //清空画布，防止缓存---------------------------------  myChart.clear();    // myChart.setOption({});//初始化数据  myChart.setOption(result);    myChart.hideLoading();  //如果不清空,图形数据将不会更改        },  error : **function**(errorMsg) {  //请求失败时执行该函数  alert("图表请求数据失败!");  }  })  }    </script>  <script type=*"text/javascript"*>  //定义列  **var** colArr = [[  {field:"id",title:"流水号",width:120,formatter:**function**(value,row,index){  **if** (row) {  **return** row.ypxx.bm;  }  }},  {field:"mc",title:"通用名",width:120,formatter:**function**(value,row,index){  **if** (row) {  **return** row.ypxx.mc;  }  }},  {field:"cgl",title:"采购量",width:120},  {field:"cgje",title:"采购金额",width:120}  ]];    //加载datagrod  $(**function**(){    $("#drugyzrealist").datagrid({  url:"${baseurl }/analyze/list\_drugyzrea.action",  title:"交易明细列表",  columns:colArr,  fitColumns:**true**,  //fixed:true,  pagination:**true**,  rownumbers:**true**,  idFiled:"ypxx.bm",  //striped:true,  });    });    //显示datagrod  **function** drugyzreaquery() {  //jquerySubByFId('yycgdqueryForm',yycgdquery\_callback);    //表单参数序列化  **var** params = $("#drugyzreaqueryForm").serializeJson();  //刷新datagride  $("#drugyzrealist").datagrid('reload',params);    }  drugyzreaquery();    $(**function**(){  $('#startDate').datebox({  onSelect: **function**(date){  **var** startDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#startDate').val(startDate);  }  });  $('#endDate').datebox({  onSelect: **function**(date){  **var** endDate = date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();  $('#endDate').val(endDate);  }  });  })    //从datagrid切换为图表  **function** tubiao2datagrid(){  //隐藏显示图表按钮  $("#btntu2data").css("display","none")  //显示datagrid按钮  $("#data2btntu").css("display","block")  //判断当前状态  //显示图表隐藏datagrid  $("#tubiao").css("display","block")  query();  $("#datagrid").css("display","none")  //修改按钮值  //$("#btntu2data").prop("iconCls","icon-database");    }    //从图表切换为datagrid  **function** datagrid2tubiao(){  $("#data2btntu").css("display","none")  $("#btntu2data").css("display","block")  drugyzreaquery();  $("#datagrid").css("display","block")  $("#tubiao").css("display","none")  }  </script> |

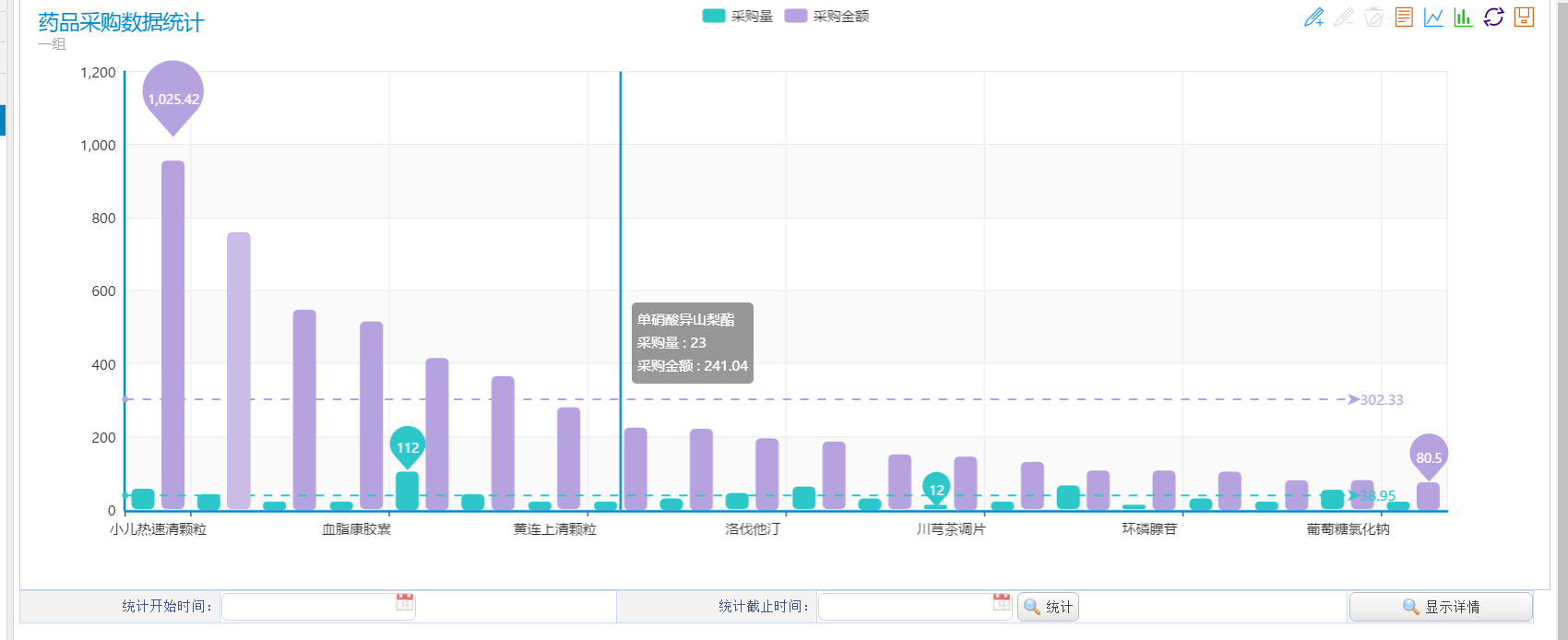
#### Action代码:

|  |
| --- |
| **public** **void** list\_drugyzrea() {  // 方法返回一个药品信息的集合  // 查询总记录数  **int** total = yybusinessService.getTotalRecordByCount(yybusinessVo.getYybusinessQueryCustom());  PageBean pageBean = **new** PageBean(yybusinessVo.getPage(), yybusinessVo.getRows(), total);  List<Yybusiness> ypxxList = yybusinessService.list\_drugyzrea(yybusinessVo.getYybusinessQueryCustom(),  pageBean.getStartIndex(), pageBean.getPageSize());  // 准备datagrid数据  write\_object(ResultUtil.*createDataGridResult*(total, ypxxList));  }  /\*\*  \* 该方法用于ajax获取药品统计数据的option  \*  \* **@throws** IOException  \*/  **public** **void** ajaxDrugyzrea\_bar() **throws** IOException {    // 创建数据承载对象  GsonOption option = **new** GsonOption();  // 初始化  List<Object> xAxisList = **new** ArrayList<>();  List<Object> barList = **new** ArrayList<>();  List<Object> barList2 = **new** ArrayList<>();  // \*\*\*\*准备数据,写死了二十条  List<Yybusiness> ypxxList = yybusinessService.list\_drugyzrea(yybusinessVo.getYybusinessQueryCustom(), 0, 20);  **if** (ypxxList!=**null**) {      **for** (Yybusiness yybusiness : ypxxList) {  xAxisList.add(yybusiness.getYpxx().getMc());  barList.add(yybusiness.getCgl());  BigDecimal cgjeBig = **new** BigDecimal(yybusiness.getCgje());  cgjeBig = cgjeBig.setScale(2, BigDecimal.***ROUND\_HALF\_UP***);  barList2.add(cgjeBig.doubleValue());  }    // \*\*\*\*\*\*设置基本属性\*\*\*\*\*\*\*start  // 设置标题和子标题(子标题即主标题下的小文字说明)  option.title().text("药品采购数据统计").subtext("一组");  // 设置鼠标移入数据柱中提示框,可自行设置  option.tooltip().trigger(Trigger.***axis***);  // 添加图标图例,每个图标仅可有一个图例(就是统计图上方的统计说明)  option.legend("采购量", "采购金额");    option.toolbox().show(**true**).feature(Tool.***mark***, Tool.***dataView***, **new** MagicType(Magic.***line***, Magic.***bar***).show(**true**),  Tool.***restore***, Tool.***saveAsImage***);  // 设置calculable值,该值是什么?不知道  option.calculable(**true**);    option.xAxis(**new** CategoryAxis().data(xAxisList.toArray()));  // yAxis:即y轴需要显示的数据,简单来说就是数据的数值,如果直接new ValueAxis()  // 应该是根据数据的需求进行自动显示  option.yAxis(**new** ValueAxis());    Bar bar = **new** Bar("采购量");  // data:就是该bar的数据,需要和上面的城镇名称进行对应    bar.data(barList.toArray());  // 设置标注,markPoint是一个标注对象,支持链式编程,可以直接设置值,很多特效可以在这里设置  bar.markPoint().data(**new** PointData().type(MarkType.***max***).name("最大值"),  **new** PointData().type(MarkType.***min***).name("最小值"));  // 设置标线,也就是横穿数据柱的一条线,代表的是平均值,详情请看MarkType对象  bar.markLine().data(**new** PointData().type(MarkType.***average***).name("平均值"));    Bar bar2 = **new** Bar("采购金额");  // data:就是该bar的数据,需要和上面的城镇名称进行对应    bar2.data(barList2.toArray());  // 设置标注,markPoint是一个标注对象,支持链式编程,可以直接设置值,很多特效可以在这里设置  bar2.markPoint().data(**new** PointData().type(MarkType.***max***).name("最大值"),  **new** PointData().type(MarkType.***min***).name("最小值"));  // 设置标线,也就是横穿数据柱的一条线,代表的是平均值,详情请看MarkType对象  bar2.markLine().data(**new** PointData().type(MarkType.***average***).name("平均值"));    option.series(bar, bar2);  }**else** {  option.setNoDataLoadingOption(**new** NoDataLoadingOption().text("无数据"));  }  getResponse().setContentType("text/html;charset=utf-8");  getResponse().getWriter().write(option.toString()); |

#### 效果图:



#### 点击显示图表效果图:



## 功能模块--根据月份统计采购金额(拓展):

### 需求分析:

统计一年中每个月的药品需求量的top20,显示为图表,卫生局或者供应商进行查看对药品进行生产量控制

#### 相关技术:

Echarts前端框架

### 代码实现:

#### Jsp页面:

|  |
| --- |
| <script type=*"text/javascript"*>      **function** query(){    // Step:3 conifg ECharts's path, link to echarts.js from current page.  // Step:3 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需图表路径  require.config({  paths: {  echarts: './js'  }  });  // Step:4 require echarts and use it in the callback.  // Step:4 动态加载echarts然后在回调函数中开始使用，注意保持按需加载结构定义图表路径  require(  [  'echarts',  'echarts/chart/bar',  'echarts/chart/line'  ],  **function**(ec) {  //--- 声明一个折柱 ---  myChart = ec.init(document.getElementById('main'), 'yizu');  //initData();//加载数据  // --- 地图 ---  /\* 读取数据时加载效果 \*/  myChart.showLoading();  initData();//加载数据    }  );  }      /\* 加载数据ajax \*/  **function** initData(){        $.ajax({  type : "post",  async : **true**, //异步请求（同步请求将会锁住浏览器，用户其他操作必须等待请求完成才可以执行）  url : "${baseurl}/analyze/MonthBar.action", //请求发送到TestServlet处  data : {},  dataType : "json", //返回数据形式为json  success : **function**(result) {  //请求成功时执行该函数内容，result即为服务器返回的json对象  //清空画布，防止缓存---------------------------------  myChart.clear();    myChart.setOption(result);    myChart.hideLoading();  //如果不清空,图形数据将不会更改        },  error : **function**(errorMsg) {  //请求失败时执行该函数  alert("图表请求数据失败!");  }  })  }  query();  </script> |

#### Action代码:

|  |
| --- |
| // 初始化  List<Object> xAxisList = **new** ArrayList<>();  List<Object> barList = **new** ArrayList<>();  // 创建数据承载对象  GsonOption option = **new** GsonOption();    List<Map<String,Object>> list = yybusinessService.findAllByMoth();    **if** (list!=**null**) {    **for** (Map<String, Object> map : list) {  xAxisList.add(map.get("YF")+"月");  BigDecimal object = (BigDecimal) map.get("CGJE");  barList.add(object.setScale(2, BigDecimal.***ROUND\_HALF\_UP***).doubleValue());  }      // \*\*\*\*\*\*设置基本属性\*\*\*\*\*\*\*start  // 设置标题和子标题(子标题即主标题下的小文字说明)  option.title().text("医药采购信息-按月份统计").subtext("一组");  // 设置鼠标移入数据柱中提示框,可自行设置  option.tooltip().trigger(Trigger.***axis***);  // 添加图标图例,每个图标仅可有一个图例(就是统计图上方的统计说明)  option.legend("采购金额");    option.toolbox().show(**true**).feature(Tool.***mark***, Tool.***dataView***, **new** MagicType(Magic.***line***, Magic.***bar***).show(**true**),  Tool.***restore***, Tool.***saveAsImage***);  // 设置calculable值,该值是什么?不知道  option.calculable(**true**);    option.xAxis(**new** CategoryAxis().data(xAxisList.toArray()));  // yAxis:即y轴需要显示的数据,简单来说就是数据的数值,如果直接new ValueAxis()  // 应该是根据数据的需求进行自动显示  option.yAxis(**new** ValueAxis());    Bar bar = **new** Bar("采购金额");  // data:就是该bar的数据,需要和上面的城镇名称进行对应    bar.data(barList.toArray());  // 设置标注,markPoint是一个标注对象,支持链式编程,可以直接设置值,很多特效可以在这里设置  bar.markPoint().data(**new** PointData().type(MarkType.***max***).name("最大值"),  **new** PointData().type(MarkType.***min***).name("最小值"));  // 设置标线,也就是横穿数据柱的一条线,代表的是平均值,详情请看MarkType对象  bar.markLine().data(**new** PointData().type(MarkType.***average***).name("平均值"));    option.series(bar);    }**else** {  option.setNoDataLoadingOption(**new** NoDataLoadingOption().text("无数据"));  }  getResponse().setContentType("text/html;charset=utf-8");  getResponse().getWriter().write(option.toString()); |

## 统计分析总结:

使用Echarts统计数据完整性便捷性可读性强,所以引用该技术