**报表组件**

## 一、综述

**曲线图、区域图、饼图、散点图、气泡图、动态图表、组合图表、3D 图、测量图、热点图、树状图(Treemap)**

## 二、 JasperReports

**1 是一个基于Java的开源报表工具，**它可以在Java环境下像其它IDE报表工具（这里我们用的是iReport对其进行可视化开发）一样来制作报表

。JasperReports支持PDF、HTML、XLS、CSV和XML文件输出格式。JasperReports是当前Java开发者最常用的报表工具。

**2 导入jasperreport相关jar包，这里采用当前比较稳定的5.2版本**

--------------------------------------------------------------------------

jasperreports-5.2.0.jar

jasperreports-applet-5.2.0.jar

jasperreports-fonts-5.2.0.jar

jasperreports-javaflow-5.2.0.jar

其他jar包：

groovy-all-2.0.1.jar： 将.jrxml编译成.jasper.如果你不使用java编译，而使用ireport工具编译则无须导入该包

iText-2.1.7.jar： 生成pdf所依赖的包

iTextAsian.jar： 支持pdf中文字体依赖的包

poi-3.7.jar： 导出excel时候用

commons-collections-2.1.1.jar

commons-digester-2.1.jar： 解析.jrxml

commons-logging-1.1.1.jar

poi-3.7-20101029.jar： 生成excel报表所依赖的jar

图形展现jfreechart相关jar包：

jfreechart-1.0.12.jar

jcommon-1.0.15.jar

**3 jasperreports与struts2集成，**则要导入struts2的jasperreports插件，可从http://mvnrepository.com/获取

------------------------------------------------------------------------------------

struts2-jasperreports-plugin-2.2.3.jar

location 是指我们刚才用iReport编译生成的jasper文件

dataSource 是指我们执行的数据库查询结果，JavaBean , List

format 是指需要输出的类型，默认是PDF，其他的输出类型有：XML、HTML、XLS、CSV、RTF ，注意这里一定要大写

除了这几个参数，还有下列参数：

delimiter 是指如果输出类型为CSV的话，指定分割符，默认为“,”

parse 是指是否解析location参数中的EL表达式，默认为 true

contentDisposition 指定disposition，默认为“inline”，如果设为“attachment”就是强制下载

documentName 输出的文件名

format 格式，默认为“PDF”，其他的有：CSV、HTML、RTF、XLS、XML

imageServletUrl 生成图形的路径

**4 struts.xml配置**

------------------------

struts2中的需继承jasperreports-default命名空间

<package name=“default” extends="struts-default, jasperreports-default">

<!-- 报表输出Action -->

<action name="JR\_\*\_\*" class="jasperReportAction">

<result name="success" type="jasper">

<param name="location">/reports/{1}.jasper</param>

<param name="connection">connection</param>

<param name="format">{2}</param>

<param name="documentName">{1}</param>

<param name="reportParameters">para</param>

<param name="imageServletUrl"><![CDATA[/image?image=]]></param>

</result>

</action>

</package>

**5 web.xml配置**

---------------------

<servlet>

<servlet-name>JasperReportImageServlet</servlet-name>

<servlet-class>net.sf.jasperreports.j2ee.servlets.ImageServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>JasperReportImageServlet</servlet-name>

<url-pattern>/image</url-pattern>

</servlet-mappin

## 三、 JFreeChart

**1 基础知识**

一个JAVA的图表引擎JFreeChart用来产生基于WEB的图表。 JFreeChart是开放源代码站点SourceForge.net上的一个JAVA项目，它主要用来各种各样的图表，这些图表包括：饼图、柱状图(普 通柱状图以及堆栈柱状图)、线图、区域图、分布图、混合图、甘特图以及一些仪表盘等等。这些不同式样的图表基本上可以满足目前的要求。

**2 开发**

导入需要的2个jar文件，jcommon-版本号.jar,jfreechart-版本号.jar。可以去官网下载：http://sourceforge.net/projects/jfreechart/files/

**（1）java后台程序**

package com.test.jfreechart;

import org.jfree.chart.ChartFactory;

import org.jfree.chart.ChartFrame;

import org.jfree.chart.JFreeChart;

import org.jfree.data.general.DefaultPieDataset;

public class JFreeChartTest

{

public static void main(String[] args)

{

DefaultPieDataset dpd=new DefaultPieDataset(); //建立一个默认的饼图

dpd.setValue("管理人员", 25); //输入数据

dpd.setValue("市场人员", 25);

dpd.setValue("开发人员", 45);

dpd.setValue("其他人员", 10);

JFreeChart chart=ChartFactory.createPieChart("某公司人员组织数据图",dpd,true,true,false);

//可以查具体的API文档,第一个参数是标题，第二个参数是一个数据集，第三个参数表示是否显示Legend，第四个参数表示是否显示提示，第五个参数表示图中是否存在URL

ChartFrame chartFrame=new ChartFrame("某公司人员组织数据图",chart);

//chart要放在Java容器组件中，ChartFrame继承自java的Jframe类。该第一个参数的数据是放在窗口左上角的，不是正中间的标题。

chartFrame.pack(); //以合适的大小展现图形

chartFrame.setVisible(true);//图形是否可见

}

}

**（2）jsp直接调用**

@web.xml

<servlet>

<servlet-name>DisplayChart</servlet-name>

<servlet-class>

org.jfree.chart.servlet.DisplayChart <!--这个固定不变-->

</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>DisplayChart</servlet-name>

<url-pattern>/DisplayChart</url-pattern>

</servlet-mapping>

@jfreeChart.jsp

<%@ page language="java" contentType="text/html; charset=GB18030"

pageEncoding="GB18030"%>

<%@ page import="org.jfree.data.general.DefaultPieDataset,org.jfree.chart.ChartFactory

,org.jfree.chart.JFreeChart,org.jfree.chart.servlet.\*" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=GB18030">

<title>Insert title here</title>

</head>

<body>

<%

DefaultPieDataset dpd = new DefaultPieDataset();

dpd.setValue("管理人员", 25);

dpd.setValue("市场人员", 25);

dpd.setValue("开发人员", 45);

dpd.setValue("其他人员", 10);

JFreeChart chart = ChartFactory.createPieChart("某公司组织结构图",dpd, true, false, false);

String fileName = ServletUtilities.saveChartAsPNG(chart,800,600,session);

//ServletUtilities是面向web开发的工具类，返回一个字符串文件名,文件名自动生成，生成好的图片会自动放在服务器（tomcat）的临时文件下（temp）

String url = request.getContextPath() + "/DisplayChart?filename=" + fileName;

//根据文件名去临时目录下寻找该图片，这里的/DisplayChart路径要与配置文件里用户自定义的<url-pattern>一致

%>

<img src="<%= url %>" width="800" height="600">

</body>

</html>

**（3）web应用和框架的使用**

struts2-jfreechart-plugin-版本号.jar到工程中。

@action类

package com.test.action;

import java.awt.Font;

import java.util.List;

import java.util.Map;

import org.jfree.chart.ChartFactory;

import org.jfree.chart.JFreeChart;

import org.jfree.chart.axis.CategoryAxis;

import org.jfree.chart.axis.CategoryLabelPositions;

import org.jfree.chart.axis.NumberAxis;

import org.jfree.chart.plot.CategoryPlot;

import org.jfree.chart.plot.PlotOrientation;

import org.jfree.chart.title.TextTitle;

import org.jfree.data.category.CategoryDataset;

import org.jfree.data.category.DefaultCategoryDataset;

import com.opensymphony.xwork2.ActionContext;

import com.opensymphony.xwork2.ActionSupport;

public class ViewResultAction extends ActionSupport

{

private JFreeChart chart;//这里变量名必须是chart，不能是其他变量名

private List<String> interest; //struts会自动类型转换，将页面传递过来的值存到List中去

public JFreeChart getChart()//getChart()方法是必须的，setChart()可以不写.

{ //在action中的chart属性的get方法中，创建chart对象，然后进行设置plot主体和颜色；以及legend颜色和字体

chart = ChartFactory.createBarChart("兴趣统计结果", "项目", "结果", this

.getDataset(), PlotOrientation.VERTICAL, false, false, false);

chart.setTitle(new TextTitle("兴趣统计结果",new Font("黑体",Font.BOLD,22)));

CategoryPlot plot = (CategoryPlot)chart.getPlot();

CategoryAxis categoryAxis = plot.getDomainAxis();

categoryAxis.setLabelFont(new Font("宋体",Font.BOLD,22));

categoryAxis.setCategoryLabelPositions(CategoryLabelPositions.UP\_45);//设置角度

NumberAxis numberAxis = (NumberAxis)plot.getRangeAxis();

numberAxis.setLabelFont(new Font("宋体",Font.BOLD,22));

return chart;

}

public List<String> getInterest()

{

return interest;

}

public void setInterest(List<String> interest)

{

this.interest = interest;

}

@Override

public String execute() throws Exception

{

return SUCCESS;

}

@SuppressWarnings("unchecked")

private void increaseResult(List<String> list)//真正在开发中是不会写在action里的，应该写在model中

{ //模拟一个临时数据库

ActionContext context = ActionContext.getContext();//struts与servlet的耦合方式一

Map map = context.getApplication();

for (String str : list)

{

if (null == map.get(str))//表示用户第一次投票

{

map.put(str, 1);

}

else

{

map.put(str, (Integer) map.get(str) + 1);

}

}

}

@SuppressWarnings("unchecked")

private CategoryDataset getDataset() //得到数据集。

{

DefaultCategoryDataset dataset = new DefaultCategoryDataset();

this.increaseResult(this.getInterest());

ActionContext context = ActionContext.getContext();

Map map = context.getApplication();

dataset.setValue((Integer) map.get("football"), "", "足球");//更新成最新值

dataset.setValue((Integer) map.get("basketball"), "", "篮球");

dataset.setValue((Integer) map.get("volleyball"), "", "排球");

dataset.setValue((Integer) map.get("badminton"), "", "羽毛球");

return dataset;

}

}

@jsp页面

<%@ page language="java" import="java.util.\*" pageEncoding="gbk"%>

<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

</head>

<body>

<h1><font color="blue">请选择喜欢的运动项目</font></h1>

<s:form action="viewResult">

<s:checkbox name="interest" label="足球" fieldValue="football"></s:checkbox>

<s:checkbox name="interest" label="篮球" fieldValue="basketball"></s:checkbox>

<s:checkbox name="interest" label="排球" fieldValue="volleyball"></s:checkbox>

<s:checkbox name="interest" label="羽毛球" fieldValue="badminton"></s:checkbox>

<!--

<s:checkboxlist list="#{'computer':'计算机','math':'数学'}" name="interest" label="课程" labelposition="top"></s:checkboxlist>

-->

<s:submit value="提交"></s:submit>

</s:form>

</body>

</html>

@struts.xml

<package name="struts2" extends="struts-default,jfreechart-default">

<action name="viewResult" class="com.test.action.ViewResultAction">

<result name="success" type="chart">

<param name="height">600</param>

<param name="width">800</param>

</result>

</action>

@struts-plugin-xml

<struts>

<package name="jfreechart-default">

<result-types>

　　<result-type name="chart" class="org.apache.struts2.dispatcher.ChartResult">

　　　　<param name="height">150</param>

　　　　<param name="width">200</param>

　　</result-type>

</result-types>

</package>

</struts>

## 四、 Highcharts

**1. 基础使用**

Highcharts 是一个用纯JavaScript编写的一个图表库。Highcharts 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

Highcharts 免费提供给个人学习、个人网站和非商业用途使用。最基本的你需要引用两个文件即可创建基础图表

<script src="http://cdn.hcharts.cn/jquery/jquery-1.8.3.min.js"></script>

<script src="http://cdn.hcharts.cn/highcharts/highcharts.js"></script>

<script src="http://cdn.hcharts.cn/highcharts/modules/exporting.js"></script>

功能模块是在 Highcharts 主要功能的基础做的扩展，是由官方发布的功能包，常用功能模块有：

更多图表类型扩展模块（highcharts-more.js）

3D 图表模块 （highcharts-3d.js）

导出功能模块（modules/exporting.js）

金字塔图表类型（modules/funnel.js）

钻取功能模块（modules/drilldown.js）

数据加载功能模块（modules/data.js）

**2 特性**

@兼容性 - 支持所有主流浏览器和移动平台（android、iOS等）。

@多设备 - 支持多种设备，如手持设备 iPhone/iPad、平板等。

@免费使用 - 开源免费。

@轻量 - highcharts.js 内核库大小只有 35KB 左右。

@配置简单 - 使用 json 格式配置

@动态 - 可以在图表生成后修改。

@多维 - 支持多维图表

@配置提示工具 - 鼠标移动到图表的某一点上有提示信息。

@时间轴 - 可以精确到毫秒。

@导出 - 表格可导出为 PDF/ PNG/ JPG / SVG 格式

@可变焦 - 选中图表部分放大，近距离观察图表；

@外部数据 - 从服务器载入动态数据。

@文字旋转 - 支持在任意方向的标签旋转。

**3 常用开发**

（1）例子1：

<html>

<head>

<meta charset="UTF-8" />

<title>77777</title>

<script src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>

<script src="http://code.highcharts.com/highcharts.js"></script>

</head>

<body>

<div id="container" style="width: 550px; height: 400px; margin: 0 auto"></div>

<script language="JavaScript">

$(document).ready(function()

{

var chart =

{

type: 'bar'

};

var title =

{

text: 'Historic World Population by Region'

};

var subtitle =

{

text: 'Source: Wikipedia.org'

};

var xAxis =

{

categories: ['Africa', 'America', 'Asia', 'Europe', 'Oceania'],

title:

{

text: null

}

};

var yAxis =

{

min: 0,

title:

{

text: 'Population (millions)',

align: 'high'

},

labels:

{

overflow: 'justify'

}

};

var tooltip =

{

valueSuffix: ' millions'

};

var plotOptions =

{

bar:

{

dataLabels:

{

enabled: true

}

}

};

var legend =

{

layout: 'vertical',

align: 'right',

verticalAlign: 'top',

x: -40,

y: 100,

floating: true,

borderWidth: 1,

backgroundColor: ((Highcharts.theme && Highcharts.theme.legendBackgroundColor) || '#FFFFFF'),

shadow: true

};

var credits =

{

enabled: false

};

var series=

[ {

name: 'Year 1800',

data: [107, 31, 635, 203, 2]

},

{

name: 'Year 1900',

data: [133, 156, 947, 408, 6]

},

{

name: 'Year 2008',

data: [973, 914, 4054, 732, 34]

}

];

var json = {};

json.chart = chart;

json.title = title;

json.subtitle = subtitle;

json.tooltip = tooltip;

json.xAxis = xAxis;

json.yAxis = yAxis;

json.series = series;

json.plotOptions = plotOptions;

json.legend = legend;

json.credits = credits;

$('#container').highcharts(json);

});

</script>

</body>

</html>

（2）例子2

<html>

<head>

<meta charset="UTF-8" />

<title>8888</title>

<script src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>

<script src="http://code.highcharts.com/highcharts.js"></script>

</head>

<body>

<div id="container" style="width: 550px; height: 400px; margin: 0 auto"></div>

<script language="JavaScript">

$(document).ready(function()

{

var chart =

{

plotBackgroundColor: null,

plotBorderWidth: null,

plotShadow: false

};

var title =

{

text: '2014 年各浏览器市场占有比例'

};

var tooltip =

{

pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'

};

var plotOptions =

{

pie:

{

allowPointSelect: true,

cursor: 'pointer',

dataLabels:

{

enabled: true,

format: '<b>{point.name}%</b>: {point.percentage:.1f} %',

style:

{

color: (Highcharts.theme && Highcharts.theme.contrastTextColor) || 'black'

}

}

}

};

var series=

[{

type: 'pie',

name: 'Browser share',

data:

[

['Firefox', 45.0],

['IE', 26.8],

{

name: 'Chrome',

y: 12.8,

sliced: true,

selected: true

},

['Safari', 8.5],

['Opera', 6.2],

['Others', 0.7]

]

}];

var json = {};

json.chart = chart;

json.title = title;

json.tooltip = tooltip;

json.series = series;

json.plotOptions = plotOptions;

$('#container').highcharts(json);

});

</script>

</body>

</html>

**4 类型对应**

区域图：var chart = {type: 'area' };

曲线区域：var chart = { type: 'areaspline' };

气泡图：var chart = {type: 'bubble', zoomType: 'xy'};

树状图：var chart = { type: 'treemap' };

测量图：var chart = { type: 'guage' };

3D图：

var chart =

{

type: 'column',

options3d:

{

enabled: true, //显示图表是否设置为3D， 我们将其设置为 true

alpha: 15, //图表视图旋转角度

beta: 15, //图表视图旋转角度

depth: 50, //图表的合计深度，默认为100

viewDistance: 25 //定义图表的浏览长度

}

};

散点图：var chart = { type: 'scatter', zoomType: 'xy' };

柱形图：var chart = {type: 'column' };

条形图：var chart = { type: 'bar'};