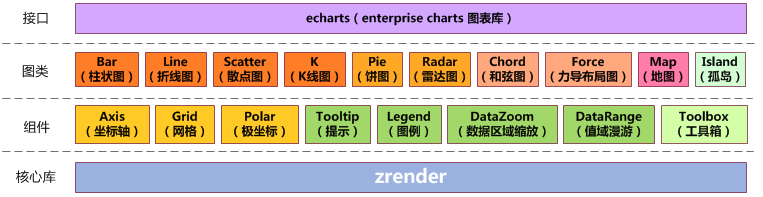
**21. 可视化**

**一、ECharts简介**

ECharts，纯Javascript图表库，基于Canvas，底层依赖ZRender，商业产品常用图表库，提供直观，生动，可交互，可个性化定制的数据可视化图表。创新的拖拽重计算、数据视图、值域漫游等特性大大增强了用户体验，赋予了用户对数据进行挖掘、整合的能力。可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等）。图表类型支持折线图（区域图）、柱状图（条状图）、散点图（气泡图）、K线图、饼图（环形图）、雷达图（填充雷达图）、和弦图、力导向布局图、地图，同时支持任意维度的堆积和多图表混合展现。

[](http://192.168.1.241/doc/exper/128/img/01.png)

**二、需求描述**

需求1：把用户对京东进行评论时使用的是客户端还是移动端的统计数据，用饼图进行数据展示，从而达到让观者能从中熟悉某个项目与整个数据组间所存在的比例关系的目的。

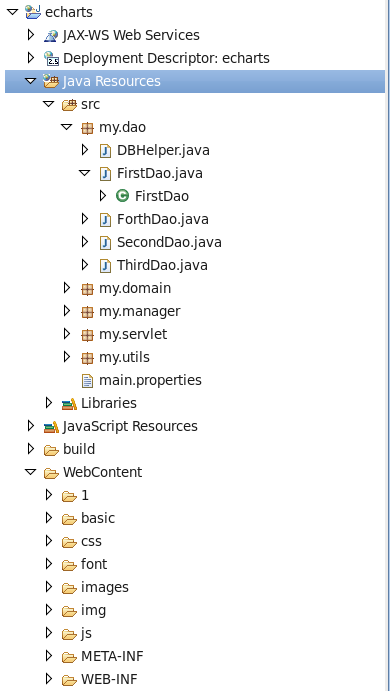
需求2：把用户在收到货后，一般多久进行评论，即用户评论周期用柱状图展示，可以达到展现数据并将数据进行比较的目的。

需求3：将购买某商品的用户级别进行统计的结果数据用饼状图展示，从而可以展现用户级别的比例构成关系，让观者能从中熟悉某个级别的用户数量与所有购买用户所存在的比例关系。

需求4：将某件商品的每天的评论量的统计数据用折线图进行展现，可以展现出这个商品每天的评论量的变化趋势。

**三、框架搭建**

1. 框架结构



2. 框架说明

框架解释说明：

(1) src目录下的五个包和一个文件是项目的后台部分，用于实现页面中对数据的操作请求

(2) WebContent下命名为1的目录存放的是用于完成可视化数据展示的jsp页面，basic目录下存放编写页面公共部分的jsp文件，js目录下为实现jsp页面所用到的js文件，WEB-INF中的lib下为项目引入的所需要的jar包和为servlet定制url的web.xml文件，其余部分主要是实现jsp页面的样式文件。

后台工作原理说明：

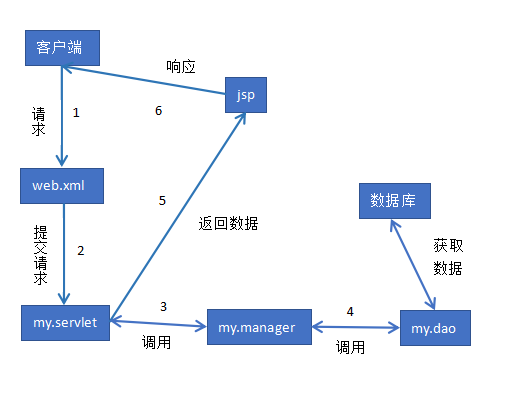
(1) 客户端向后台发送请求。即在jsp页面加载数据部分，设置需要调用的servlet文件名称和调用的方法参数给后台，目的是告诉后台要调用哪个servlet文件和这个文件中的哪个方法。

(2) web.xml文件里面设置了各个servlet文件的路径，所以客户端发送过来的请求会通过web.xml，提交到相应的如图（框架结构图）my.servlet目录下的servlet类，并接收传过来的参数。

(3) 当成功提交到这个servlet类时，再通过参数值确定使用servlet哪个方法，然会通过如图（框架结构图）my.manager目录下相应的类里面的方法，将获取的数据的值返回。

(4) my.manager中的方法在获取数据时，需要图中（框架结构图）my.dao目录下相应的类里面的方法去操作数据库，因为dao层与db层结合可以直接操作数据库，获取数据库中的数据。

工作原理如图所示：

[](http://192.168.1.241/doc/exper/128/img/03.png)

**四、实验步骤**

1. 数据包和程序包：创建/home/hfut/keshihua-data文件夹。将edu5out.sql和echarts.tar.gz放到文件夹中。

[hfut@master ~]$ mkdir -p /home/hfut/keshihua-data

[hfut@master ~]$ cp ~/resources/keshihua-data/edu5out.sql ~/keshihua-data/

[hfut@master ~]$ cp ~/resources/keshihua-data/echarts.tar ~/keshihua-data/

2. 数据库端准备工作

查看MySQL服务是否已经启动

[hfut@master ~]$ su root

[root@master hfut]# service mysqld status

如果MySQL未启动，则启动MySQL

[root@master hfut]# service mysqld start

使用root用户进入MySQL Shell

[root@master hfut]# mysql -u root

创建并使用edu5out数据库

mysql> create database edu5out;

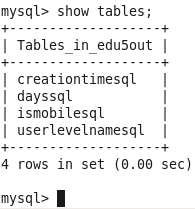
mysql> use edu5out;

将/home/hfut/keshihua-data目录下的edu5out.sql脚本导入到mysql的edu5out库中

mysql> source /home/hfut/keshihua-data/edu5out.sql;

查看一下edu5out库下的所有表

mysql> show tables;



3. 安装tomcat

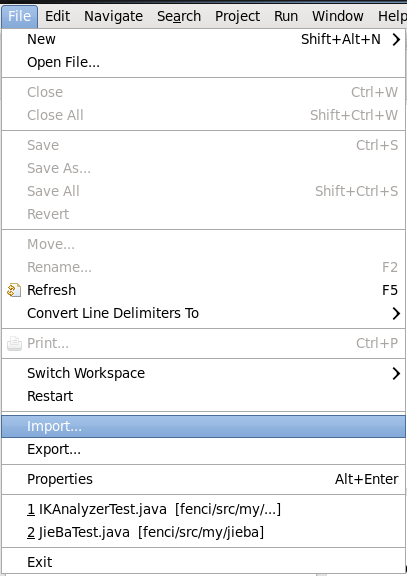
[hfut@master ~]$ wget https://archive.apache.org/dist/tomcat/tomcat-6/v6.0.53/bin/apache-tomcat-6.0.53.tar.gz

[hfut@master ~]$ tar -zxvf apache-tomcat-6.0.53.tar.gz

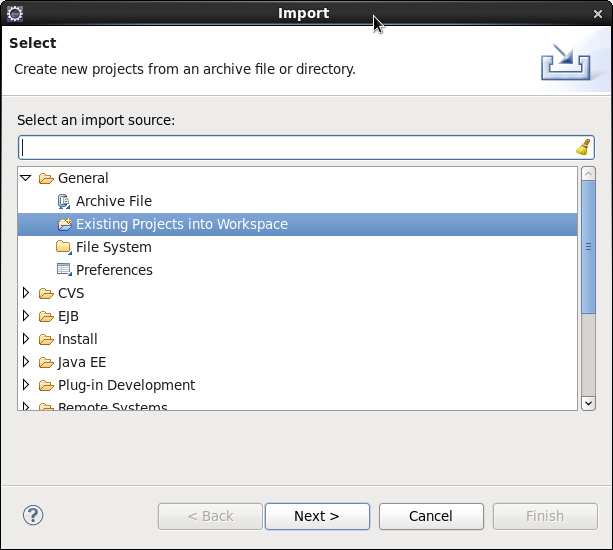
[hfut@master ~]$ mv apache-tomcat-6.0.53 tomcat

4. 导入echarts项目

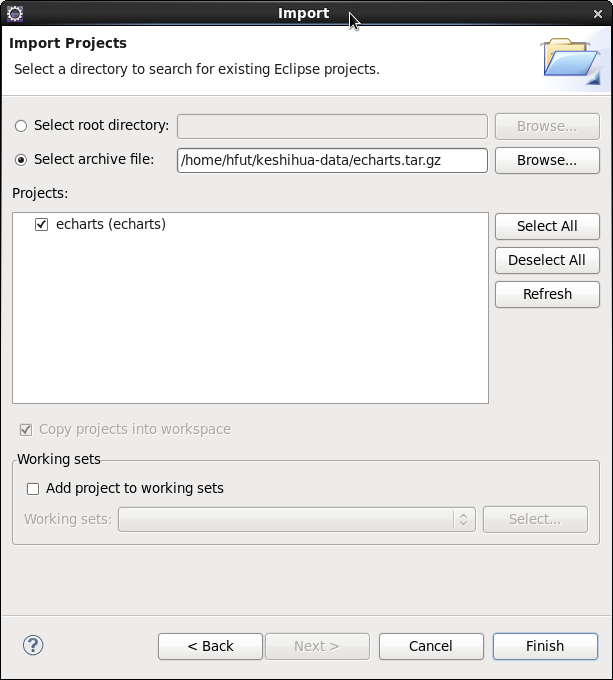
打开Eclipse，单击File选择Import。



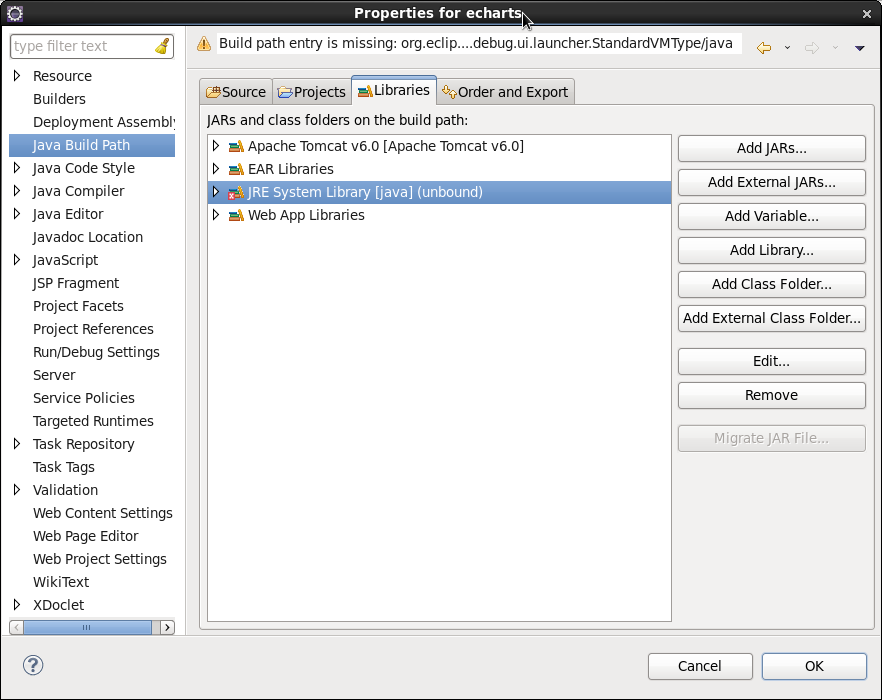
选择General中的Existing Projects into Workspace



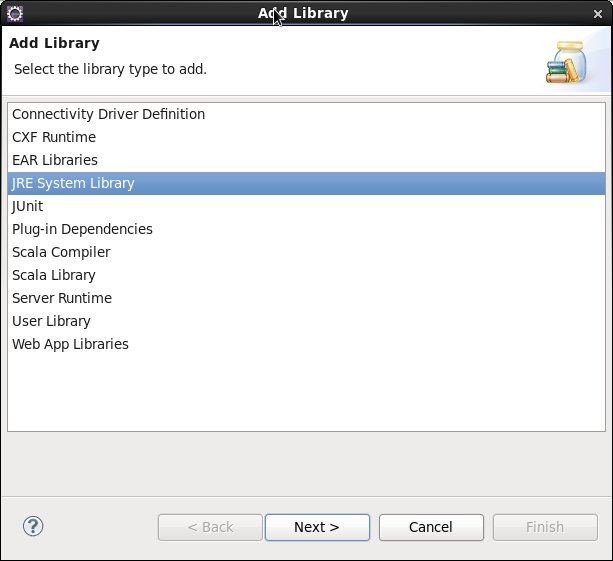
选择/home/hfut/keshihua-data/echarts.tar.gz

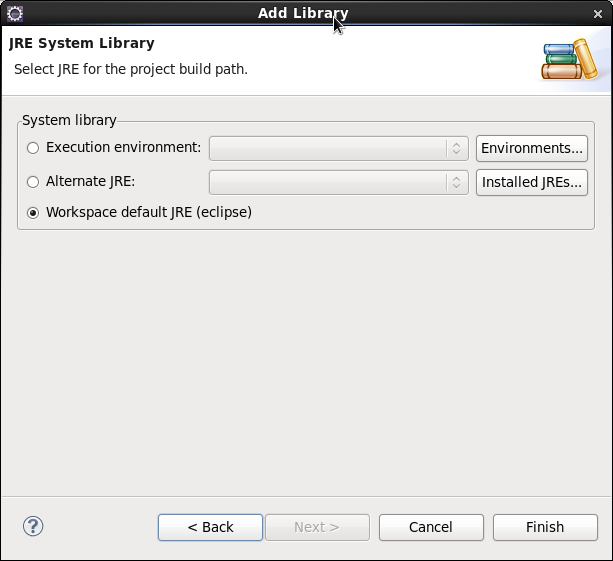


将echarts 项目的JRE System Library替换为正确的配置，点击Remove删除错误的配置。

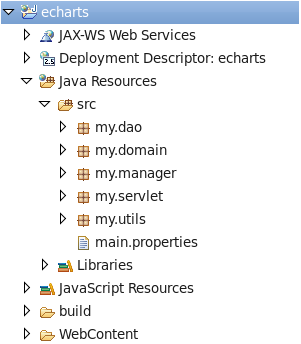


点击Add Library…添加正确的配置

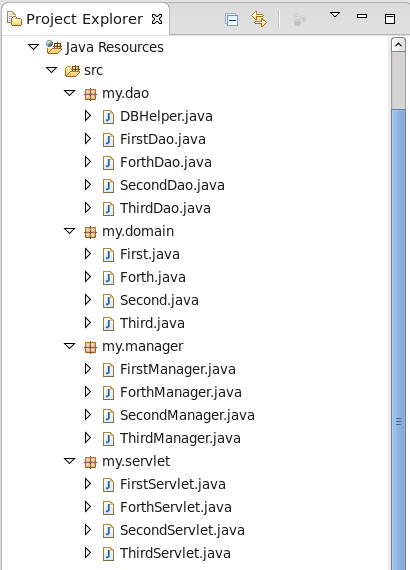


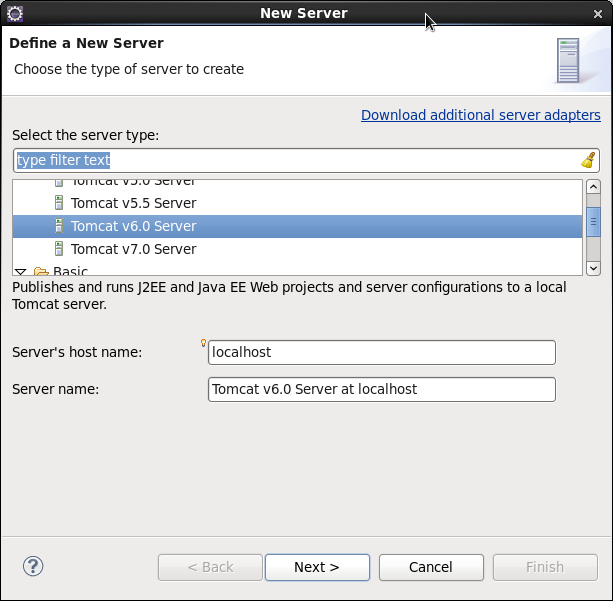


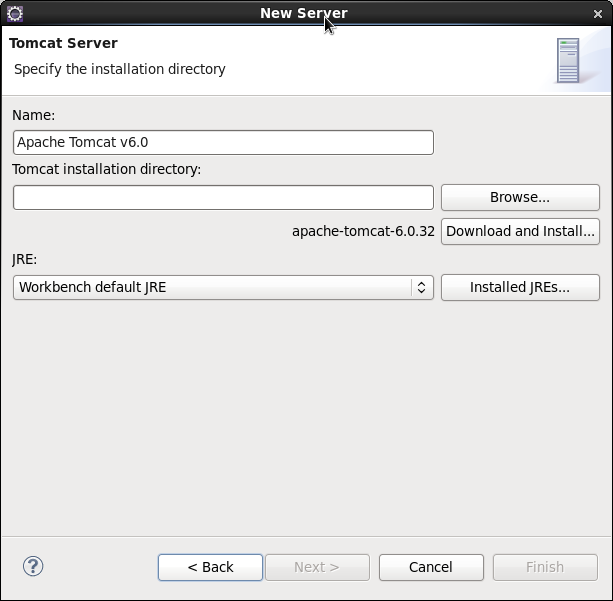
这样就将echarts项目导入进来了，目录结构如下：

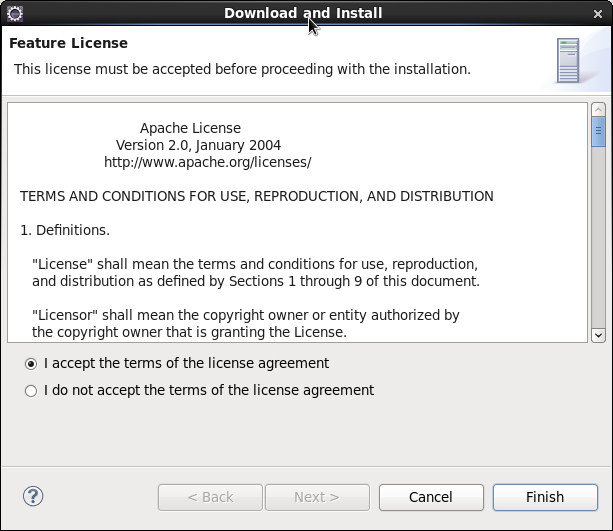


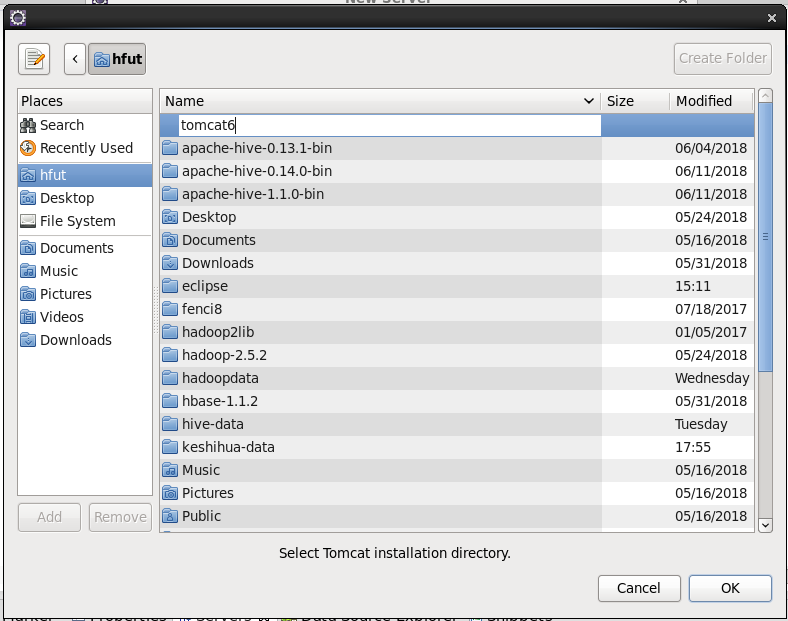
在每个包下都包含First、Second、Third、Forth四个类，分别对应四个需求的功能实现

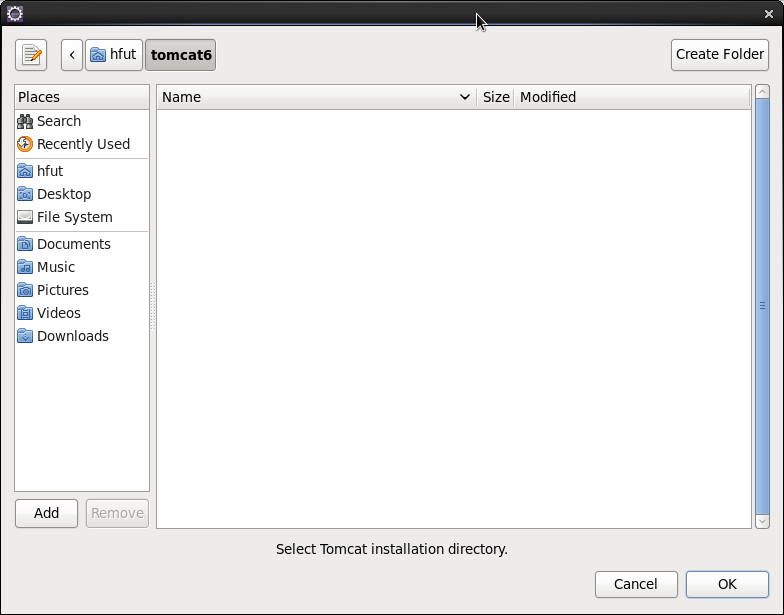


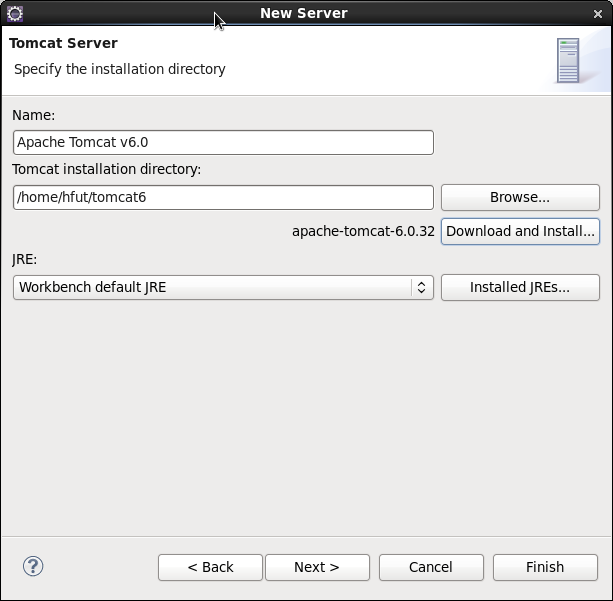








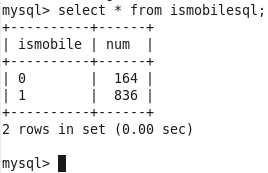




5. 需求一：将用户使用移动端还是PC端的统计结果，通过Echarts饼图展示出来。

首先，我们查看一下MySQL数据库中的ismobilesql表

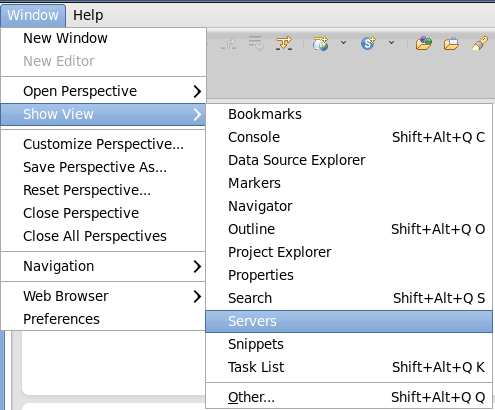
mysql> select \* from ismobilesql;



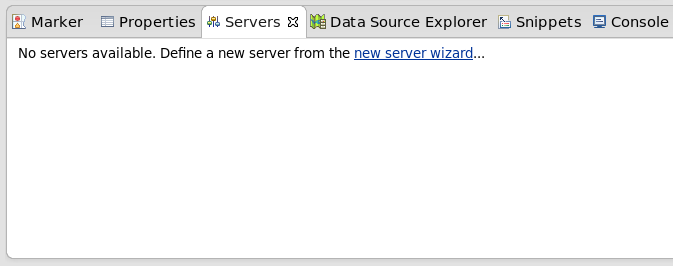
包含两个字段ismobile（是否是移动端）和num（数量），其中1代表移动端，0代表PC端。

我们来执行echarts项目，查看一下效果

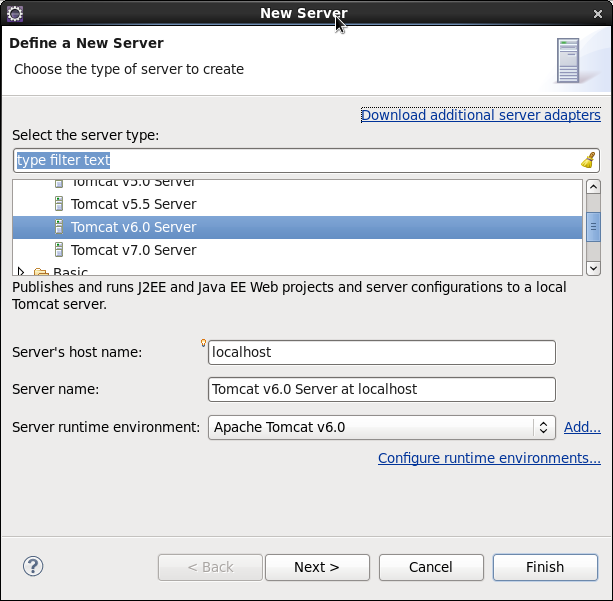
选择Window->Show View->Other



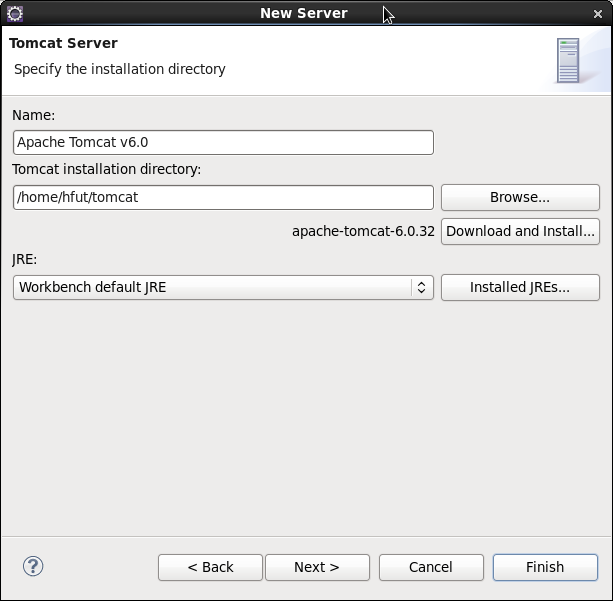
点击new server wizard…新建tomcat服务器：



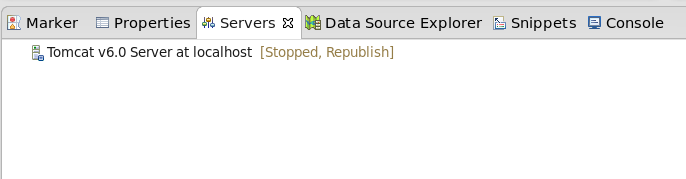
选择Tomcat v6.0 Server



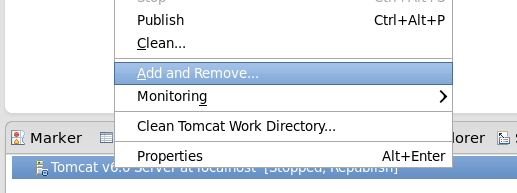
Tomcat installation directory填/home/hfut/tomcat



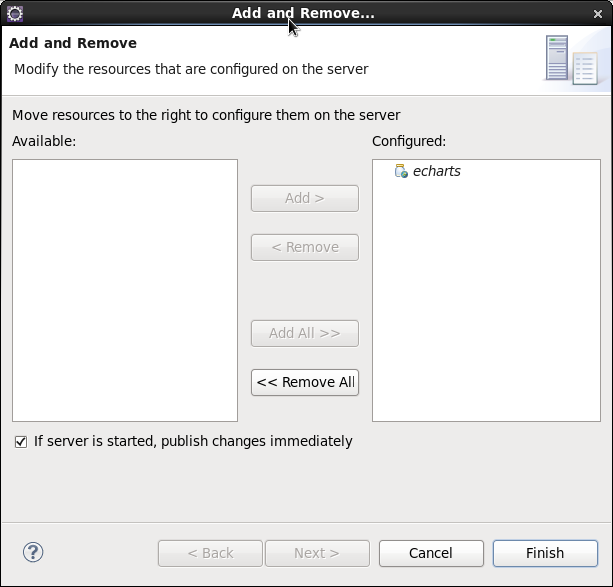
点击Finish完成tomcat配置



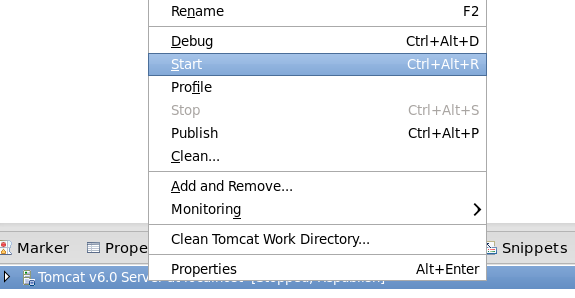
在Server处右键选择Add and Remove



将echarts项目添加到Configured中后，点击Finish

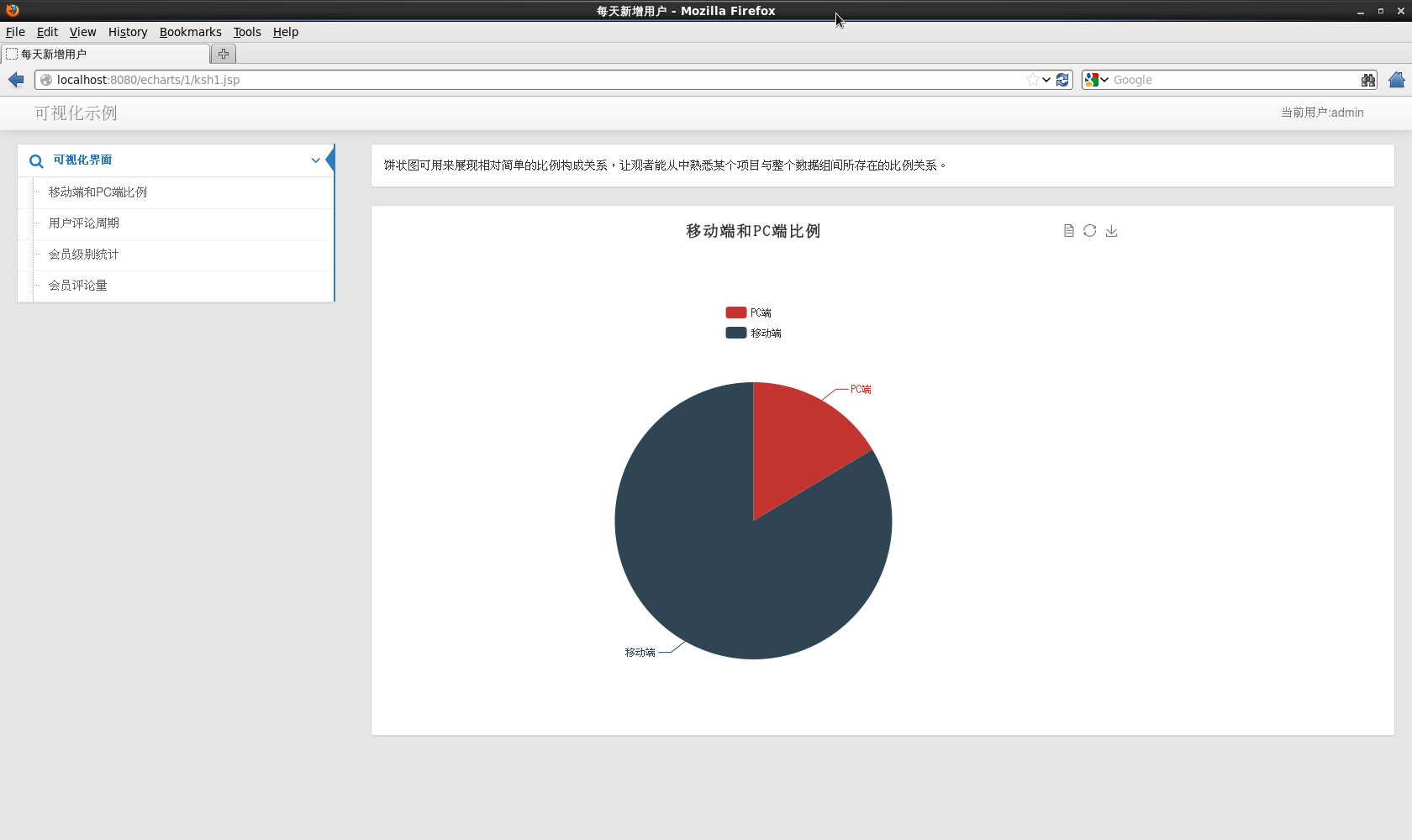


右键Server选择Start，开启Tomcat



Tomcat开启后，打开Firefox浏览器，输入下面网址：

http://localhost:8080/echarts/1/ksh1.jsp



可以看到移动端和PC端的比例结果已经通过饼状图展现出来了。

下面我们来看一下具体的代码：

在my.domain包下，First类的作用为：ismobile和num对象，定义其属性。具体代码如下：

package my.domain;

public class First {

private String ismobile;

private int num;

public First() {

}

public String getIsmobile() {

return ismobile;

}

public void setIsmobile(String ismobile) {

this.ismobile = ismobile;

}

public int getNum() {

return num;

}

public void setNum(int num) {

this.num = num;

}

@Override

public String toString() {

return "First [ismobile=" + ismobile + ", num=" + num + "]";

}

public First(String ismobile, int num) {

super();

this.ismobile = ismobile;

this.num = num;

}

}

在my.dao包下，FirstDao.java类的作用为：调用sql语句对数据库的数据进行具体操作。具体代码如下：

package my.dao;import java.sql.Connection;import java.sql.ResultSet;import java.sql.Statement;import java.util.ArrayList;import java.util.List;import my.domain.First;import org.apache.log4j.Logger;

public class FirstDao {public static Logger logger = Logger.getLogger(FirstDao.class);

public static List<First> getFirstList() { String sql = "select num,case when ismobile=1 then '移动端' when ismobile=0 then 'PC端' end as ismobile, count(1) as num from ismobilesql group by num"; Connection conn= null; ResultSet set = null; Statement stmt = null; List<First> list = new ArrayList<First>(); try { conn = DBHelper.connDB(); stmt = conn.createStatement(); set = stmt.executeQuery(sql); while (set.next()) { First bean = new First(); bean.setIsmobile(set.getString("ismobile")); bean.setNum(set.getInt("num")); list.add(bean); } } catch (Exception e) { logger.error(e.getMessage()); } finally{ DBHelper.free(set, stmt, conn); } return list;}

}在my.manager包下，FirstManager.java类的作用为：用以调用FirstDao中的方法得到数据。具体代码如下：

**package** my.manager;

**import** java.util.List;

**import** my.dao.FirstDao;

**import** my.domain.First;

**public** **class** FirstManager {

**public** List<First> getFirstList() {

**return** FirstDao.*getFirstList*();

}

}

在my.servlet包下，FirstServlet.java类的作用为：将前面得到的数据传给jsp页面，响应客户端。具体代码如下：

**package** my.servlet;

**import** java.io.PrintWriter;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** org.apache.log4j.Logger;

**import** my.domain.First;

**import** my.manager.FirstManager;

**import** my.utils.BaseServlet;

**import** net.sf.json.JSONArray;

**public** **class** FirstServlet **extends** BaseServlet {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**public** **static** Logger *logger* = Logger.*getLogger*(FirstServlet.**class**);

@Override

**public** **void** execute(HttpServletRequest request, HttpServletResponse response)

**throws** Exception {

response.setContentType("text/html;charset=utf-8");

String opt = request.getParameter("opt");

FirstManager scale = **new** FirstManager();

**if** (opt.equals("First")) {

List<First> list = **new** ArrayList<First>();

list = scale.getFirstList();

JSONArray jsonArray = JSONArray.*fromObject*(list);

// System.err.println(jsonArray);

PrintWriter out = response.getWriter();

out.print(jsonArray);

out.flush();

out.close();

}

}

}

main.properties文件的作用为，配置MySQL相关参数，涉及用户名、密码、数据库等。具体代码如下：

##MySQL Configuration

JDBC.DRIVER=com.mysql.jdbc.Driver

JDBC.URL=jdbc:MySql://localhost:3306/edu5out?zeroDateTimeBehavior=convertToNull&characterEncoding=utf8

MYSQLDB.USER=root

MYSQLDB.PASSWORD=

web.xml的作用为：接受客户端的请求，然后响应给相应的servlet类，就好比一个信息中转站。主要代码为：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" id="WebApp\_ID" version="2.5">

<display-name></display-name>

<servlet>

<servlet-name>FirstServlet</servlet-name>

<servlet-class>my.servlet.FirstServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>ForthServlet</servlet-name>

<servlet-class>my.servlet.ForthServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>ThirdServlet</servlet-name>

<servlet-class>my.servlet.ThirdServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>SecondServlet</servlet-name>

<servlet-class>my.servlet.SecondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>FirstServlet</servlet-name>

<url-pattern>/FirstServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>ForthServlet</servlet-name>

<url-pattern>/ForthServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>ThirdServlet</servlet-name>

<url-pattern>/ThirdServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>SecondServlet</servlet-name>

<url-pattern>/SecondServlet</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

</web-app>

在WebContent/1/下面新建一个jsp页面，命名为ksh1，这个页面就是在浏览器上展现给客户的页面，在这个页面上接收后台数据，主要作用是实现数据的可视化展现。具体代码如下：

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html lang=*"zh-CN"*>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<meta charset=*"utf-8"*>

<title>每天新增用户</title>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<%@include file=*"../basic/cssjs.jsp"* %>

<!-- 引入Jquery包 -->

<script type=*"text/javascript"* src=*"../js/jquery-2.1.4.min.js"*></script>

<!-- 引入Echarts3包 -->

<script type=*"text/javascript"* src=*"../js/echarts.min.js"*></script>

</head>

<body>

<%@include file=*"../basic/top.jsp"* %>

<!-- content start -->

<div class=*"container-fluid"*>

<div class=*"row-fluid"*>

<div class=*"span3"*>

<!-- left navbar start -->

<%@include file=*"../basic/left.jsp"* %>

<!-- left navbar end -->

</div>

<!-- right content start -->

<div class=*"span9"*>

<div class=*"session well"*>饼状图可用来展现相对简单的比例构成关系，让观者能从中熟悉某个项目与整个数据组间所存在的比例关系。</div>

<div class=*"session"*>

<div id=*"main"* style="width: *100%*;height:*600px*;"></div>

</div>

</div>

<!-- right content end -->

</div>

</div>

<!-- content end -->

</body>

</html>

<script type=*"text/javascript"*>

// 基于准备好的dom，初始化echarts实例

**var** myChart = echarts.init(document.getElementById('main'));

// 指定图表的配置项和数据

myChart.setOption({

title : {

text: '移动端和PC端比例',

x:'center'

},

tooltip : {

trigger: 'item',

formatter: '{a} <br/>{b} : {c} ({d}%)'

},

legend: {

orient: 'vertical',

y:'100',

data: []

},

toolbox: {

show : **true**,

feature : {

mark : {show: **true**},

dataView : {show: **true**, readOnly: **false**},

restore : {show: **true**},

saveAsImage : {show: **true**}

}

},

series : [

{

name: '移动端和PC端比例',

type: 'pie',

radius : '55%',

center: ['50%', '60%'],

//data:[{}],

data:[],

itemStyle: {

emphasis: {

shadowBlur: 10,

shadowOffsetX: 0,

shadowColor: 'rgba(0, 0, 0, 0.5)'

}

}

}

]

});

//异步加载数据

**var** mapOnlyKey = [];

**var** mapKeyValue = [];

**var** mapOnlyValue = [];

**var** info = {"opt":"First"};

$.post("../FirstServlet", info, **function**(data){

mapOnlyKey.length=0;

mapKeyValue.length=0;

mapOnlyValue.length=0;

**for**(**var** i=0; i < data.length; i++){

mapOnlyKey.push( data[i].ismobile);

mapKeyValue.push({"value":Math.round(data[i].num), "name": data[i].ismobile });

mapOnlyValue.push( data[i].num );

}

//console.log(mapOnlyKey);

//console.log(mapKeyValue);

// console.log(mapOnlyValue);

//return false;

// 填入数据

myChart.setOption({

legend: {

//类别

data: mapOnlyKey

},

series: [{

// 根据名字对应到相应的系列

name: '数量',

data:mapKeyValue

}]

});

// 使用刚指定的配置项和数据显示图表。

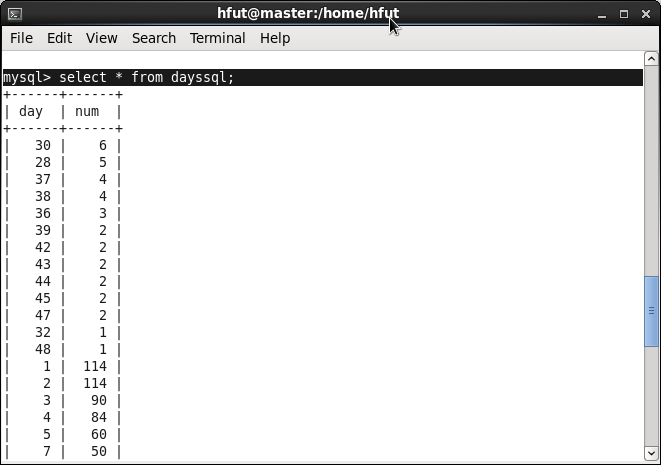
}, 'json');

</script>

6. 需求二：将用户收到货后多久进行评价的统计结果，通过Echarts柱状图展示出来。

我们查看一下MySQL数据库中的dayssql表

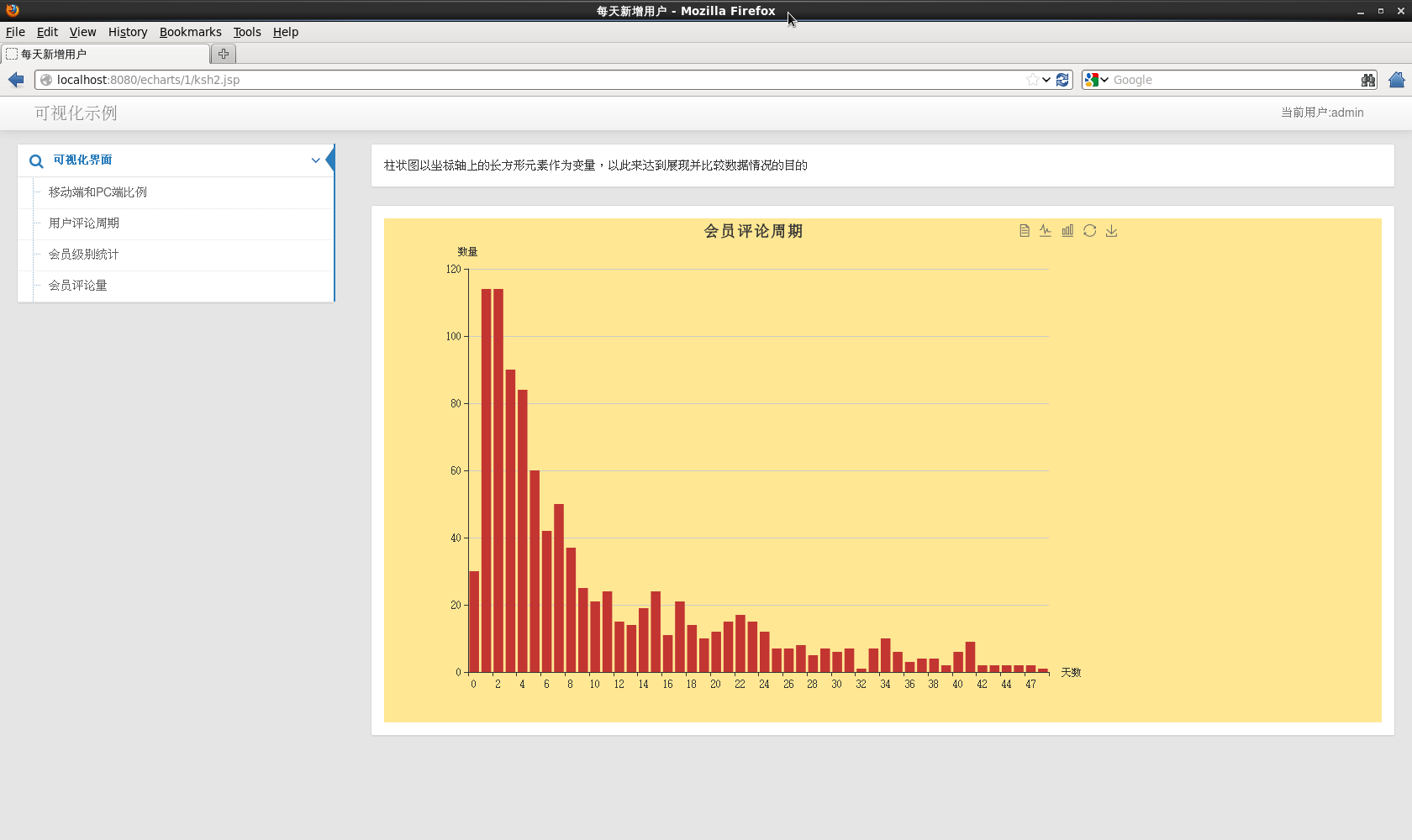
mysql> select \* from dayssql;



包含day（天数）和num（数量）两个字段。

打开Firefox浏览器，输入下面网址：

http://localhost:8080/echarts/1/ksh2.jsp



可以看到会员评论周期的数据通过柱状图清晰的展现出来了。

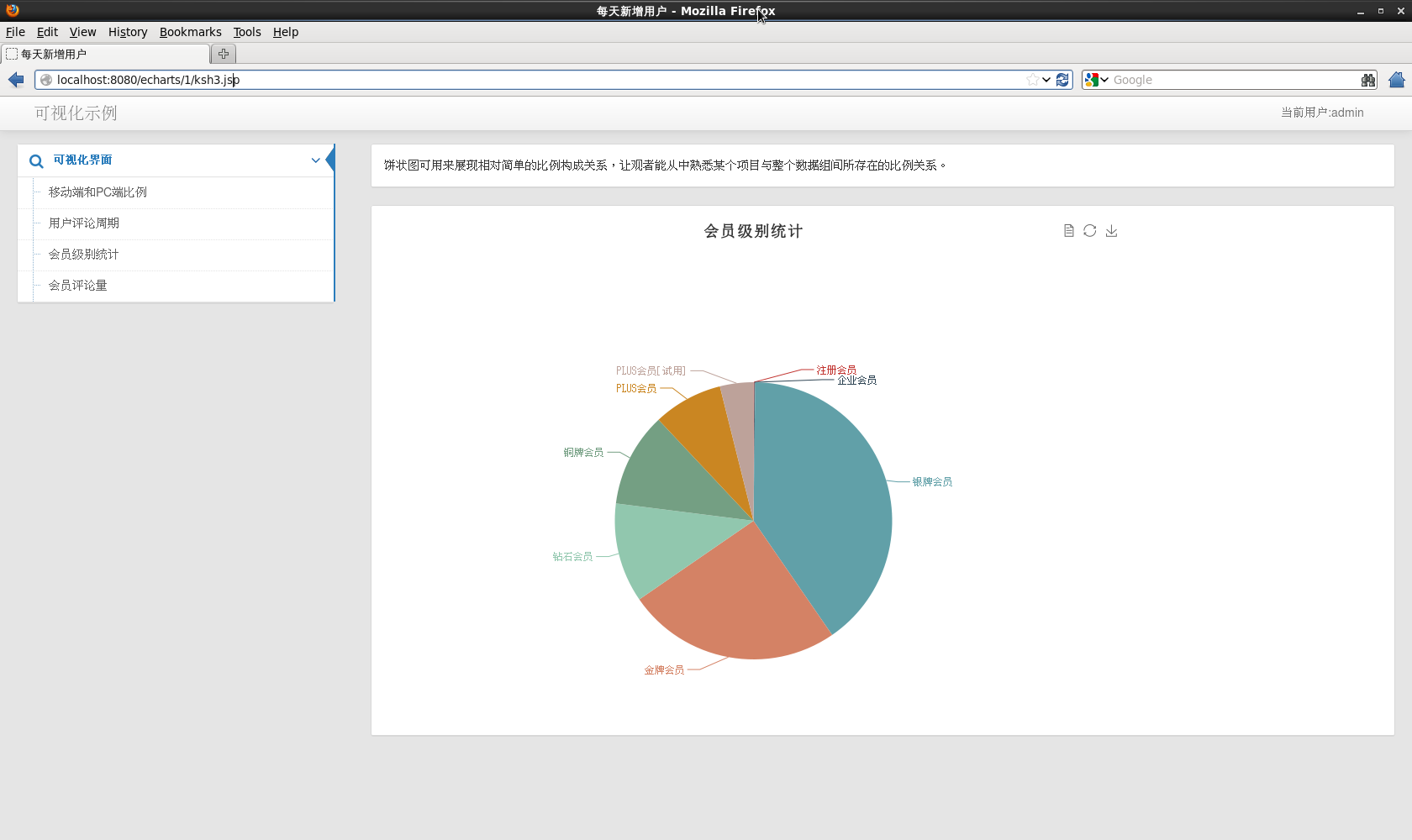
7. 需求三：将购买该商品的用户级别统计结果，通过Echarts饼状图展示出来。

我们查看一下MySQL数据库中的userlevelnamesql表

mysql> select \* from userlevelnamesql;

打开Firefox浏览器，输入下面网址：

http://localhost:8080/echarts/1/ksh3.jsp



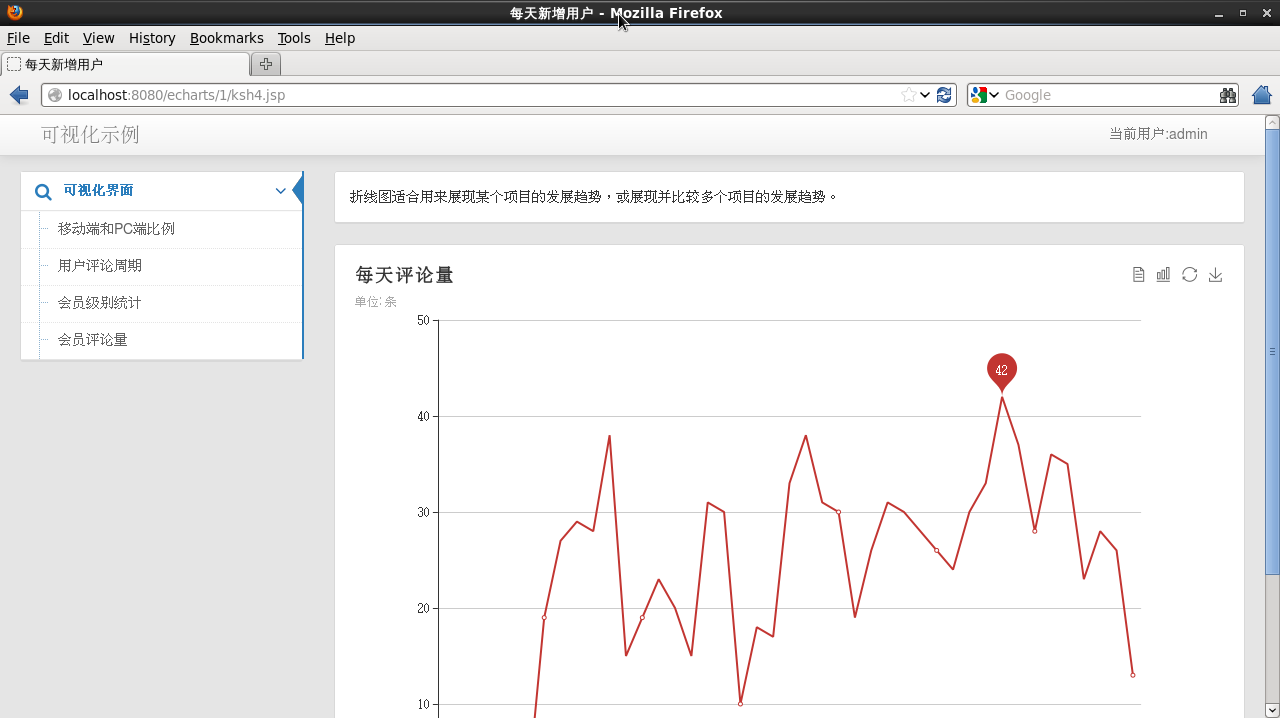
8. 需求四：将该商品每天评论量的统计结果，通过Echarts折线图展示出来。

我们查看一下MySQL数据库中的creationtimesql表：

mysql> select \* from creationtimesql;

打开Firefox浏览器，输入下面网址：

http://localhost:8080/echarts/1/ksh4.jsp



可以看到该商品每天评论量的统计结果，通过折线图清晰的展现出来了。