# 爬虫比价器

**内参保密不得外传**

**达内Java培优方向**

**陈子枢 主讲**

**QQ: 1920-9520-78**

**http://dobetter.tedu.cn**

**2017**

# 第一天：爬取新闻网站

## 比价器系统

### 比价系统功能

* 利用Jsoup爬取每个页面的商品信息
* ECharts柱状图、曲线图

### 工作原理

利用每个电商网站的搜索条查询同样的条件，例如：iphoneX，然后发现其规律

https://search.gome.com.cn/search?question=iphonex%2064g&searchType=goods

https://search.suning.com/iphonex%2064g/

https://list.tmall.com/search\_product.htm?q=iphonex+64g

通过url传递了查询条件。

**抓取步骤：**

1. 用户输入查询条件
2. 分别到每个电商网站访问，利用其查询出它们网站的对应商品
3. 获取其列表页面中第一个商品的链接
4. 利用jsoup爬取每个页面的商品信息
5. 把商品信息入库并设置爬取时间
6. 使用ECharts进行进行价格的比较
7. 拓展：ECharts实现抓取价格柱状图、饼形图、曲线图

## 涉及的技术点

### JavaScript

JavaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。

在1995年时，由Netscape公司的Brendan Eich，在网景导航者浏览器上首次设计实现而成。因为Netscape与Sun合作，Netscape管理层希望它外观看起来像Java，因此取名为JavaScript。

### json

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

### HttpClient

HttpClient是Apache Jakarta Common下的子项目，可以用来提供高效的、最新的、功能丰富的支持HTTP协议的客户端编程工具包，并且它支持HTTP协议最新的版本和建议。

注意：不要随意升级版本，每次大版本内容api变化比较大。

### Jsoup

jsoup 是一款Java 的HTML解析器，可直接解析某个URL地址、HTML文本内容。它提供了一套非常省力的API，可通过DOM，CSS以及类似于jQuery的操作方法来取出和操作数据。

## 爬虫新闻

### Httpclient

package test;

import java.io.IOException;

import org.apache.http.HttpEntity;

import org.apache.http.client.methods.CloseableHttpResponse;

import org.apache.http.client.methods.HttpGet;

import org.apache.http.impl.client.CloseableHttpClient;

import org.apache.http.impl.client.HttpClientBuilder;

import org.apache.http.impl.client.HttpClients;

import org.apache.http.util.EntityUtils;

import org.junit.Test;

public class TestPrice {

@Test

public void taobaoItemPrice() throws IOException{

String itemId = "560563554738";

String url = "http://mdskip.taobao.com/core/initItemDetail.htm?isRegionLevel=true&itemTags=385,775,843,1035,1163,1227,1478,1483,1539,1611,1863,1867,1923,2049,2059,2242,2251,2315,2507,2635,3595,3974,4166,4299,4555,4811,5259,5323,5515,6145,6785,7809,9153,11265,12353,12609,13697,13953,16321,16513,17473,17537,17665,17857,18945,19841,20289,21762,21826,25922,28802,53954&tgTag=false&addressLevel=4&isAreaSell=false&sellerPreview=false&offlineShop=false&showShopProm=false&isIFC=false&service3C=true&isSecKill=false&isForbidBuyItem=false&cartEnable=true&sellerUserTag=839979040&queryMemberRight=true&itemId="+itemId+"&sellerUserTag2=306250462070310924&household=false&isApparel=false¬AllowOriginPrice=false&tmallBuySupport=true&sellerUserTag3=144467169269284992&sellerUserTag4=1152930305168967075&progressiveSupport=true&isUseInventoryCenter=false&tryBeforeBuy=false&callback=setMdskip×tamp=1420351892310";

HttpClientBuilder builder = HttpClients.custom();

builder.setUserAgent("Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:0.9.4)");

CloseableHttpClient httpClient = builder.build();

final HttpGet httpGet = new HttpGet(url);

httpGet.addHeader("Referer", "http://detail.tmall.com/item.htm?id="+itemId+"&skuId=68347779144&areaId=110000&cat\_id=50024400&rn=763d147479ecdc17c2632a4219ce96b3&standard=1&user\_id=263726286&is\_b=1");

CloseableHttpResponse response = null;

response = httpClient.execute(httpGet);

final HttpEntity entity = response.getEntity();

String result = null;

if (entity != null) {

result = EntityUtils.toString(entity);

EntityUtils.consume(entity);

}

//商品价格的返回值，需要解析出来价格

result = result.substring(10, result.length()-1);

}

}

### Jsoup

package test;

import java.io.IOException;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

import org.junit.Test;

public class TestNew {

@Test

public void site() throws IOException{

String url = "http://ent.qq.com/a/20171117/007399.htm";

String html = Jsoup.connect(url).execute().body();

System.out.println(html);

}

@Test

public void title() throws IOException{

String url = "http://ent.qq.com/a/20171117/007399.htm";

Document doc = Jsoup.connect(url).get();

Elements els = doc.select(".hd h1");

Element ele = els.get(0);

String title = ele.text();

System.out.println(title);

}

@Test

public void img() throws IOException{

String url = "http://ent.qq.com/a/20171117/007399.htm";

String imageUrl = Jsoup.connect(url).get()

.select(".Cnt-Main-Article-QQ p img")

.get(0)

.attr("src");

System.out.println(imageUrl);

}

}

### 抓取价格 – json

2017年4月，京东开始对价格进行反爬虫控制，访问过多的IP地址会被禁止。

@Test //价格

public void getItemPrice() throws IOException{

String url = "http://p.3.cn/prices/mgets?skuIds=J\_3882469";

String json = Jsoup.connect(url).ignoreContentType(true).execute().body();

JsonNode jsonNode = MAPPER.readTree(json);

//解析完数组，获取数组第一条数据，获取它的p元素值

Double price = jsonNode.get(0).get("p").asDouble();

log.debug(price);

}

### 抓取描述 - jsonp

@Test //商品描述

public void getItemDesc() throws IOException{

String url = "http://d.3.cn/desc/3882469";

String jsonp = Jsoup.connect(url).ignoreContentType(true).execute().body();

String json = jsonp.substring(9, jsonp.length()-1); //把函数名去掉

JsonNode jsonNode = MAPPER.readTree(json);

String desc = jsonNode.get("content").asText();

log.debug(desc);

}

# 第二天：爬取京东商城

## 抓取的五种方式

### 抓取页面

@Test //抓整个页面

public void html() throws IOException{

String url = "http://tech.qq.com/a/20170330/003855.htm";

//doc代表一个页面

String html = Jsoup.connect(url).execute().body();

System.out.println(html);

}

### 抓取整个网站

@Test //抓整站，找到所有a链接，然后进行广度优先/深度优先进行遍历

public void getAllATag() throws IOException{

String url = "http://tech.qq.com/a/20170330/003855.htm";

//获取到页面

Document doc = Jsoup.connect(url).get();

//获取到页面中的所有a标签

Elements eles = doc.getElementsByTag("a");

for(Element ele : eles){

String title = ele.text(); //获取a标签的内容

String aurl = ele.attr("href"); //获取a标签的属性

log.debug(title+" - "+aurl);

}

}

### 抓取标题 - 页面上的内容

可以多级父子样式嵌套

@Test //京东商城，商品标题

public void getItemTile() throws IOException{

String url = "https://item.jd.com/3882469.html";

Document doc = Jsoup.connect(url).get();

Element ele = doc.select(".itemInfo-wrap .sku-name").get(0);

String title = ele.text();

log.debug(title);

}

@Test //当当商城，商品标题

public void getDDItemTile() throws IOException{

String url = "http://product.dangdang.com/1052875306.html";

Document doc = Jsoup.connect(url).get();

Element ele = doc.select("article").get(0);

String title = ele.text();

log.debug(title);

}

### 抓取价格 – json

2017年4月，京东开始对价格进行反爬虫控制，访问过多的IP地址会被禁止。

@Test //价格

public void getItemPrice() throws IOException{

String url = "http://p.3.cn/prices/mgets?skuIds=J\_3882469";

String json = Jsoup.connect(url).ignoreContentType(true).execute().body();

JsonNode jsonNode = MAPPER.readTree(json);

//解析完数组，获取数组第一条数据，获取它的p元素值

Double price = jsonNode.get(0).get("p").asDouble();

log.debug(price);

}

### 抓取描述 - jsonp

@Test //商品描述

public void getItemDesc() throws IOException{

String url = "http://d.3.cn/desc/3882469";

String jsonp = Jsoup.connect(url).ignoreContentType(true).execute().body();

String json = jsonp.substring(9, jsonp.length()-1); //把函数名去掉

JsonNode jsonNode = MAPPER.readTree(json);

String desc = jsonNode.get("content").asText();

log.debug(desc);

}

## 爬取京东

抓取商品先要找到商品ID，有两个方案：

方案一：商品ID是一串数字，猜测它是自增的，于是我们可以是做一个自增的循环。但如果商品的ID不是连续，会造成很多访问无法继续访问，报链接超时。

方案二：找到网站的所有商品的列表页面，解析html找到商品的ID，这个方式解析麻烦些，但商品ID直接可以获得。

所有一般来说都是采用第二种方案。

**分类、商品列表、商品详情**

那抓取京东网站就变成抓取所有分类，按分类找到商品列表页面，从商品列表页面抓取出商品ID，最终循环商品ID，抓取所有商品详情页面，解析商品详情页面，找到所有商品的详细信息。

**断点抓取、离线分析**

京东有近22个大类143个二级分类,1286三级分类，8615683种商品，近九百万种商品。如果持续在线抓取，会很快比屏蔽。也不方便测试。所以我们采取断点抓取，离线分析。先将分类抓取，将榨取后的信息保存到磁盘中，后期对磁盘中的文件进行分析入库。

### 抓取商品分类

@Test //抓取商品分类（22,143,1286），http://www.jd.com/allSort.aspx

public void itemCat() throws IOException{

String url = "http://www.jd.com/allSort.aspx";

Document doc = Jsoup.connect(url).get();

Elements level1 = doc.select("h2 span");

log.info("大类总数："+level1.size());

for(Element ele : level1){

log.info(ele.text());

}

Elements level2 = doc.select("dl dt a");

log.info("二级分类总数："+level2.size());

for(Element ele : level2){

log.info(ele.text());

}

Elements level3 = doc.select("dl dd a");

log.info("三级分类总数："+level3.size());

for(Element ele : level3){

log.info(ele.text()+" "+ele.attr("href"));

}

}

@Test //抓取某个分类下的商品数

public void itemCatCountOne() throws IOException{

String url = "http://list.jd.com/list.html?cat=9987,653,655";

Document doc = Jsoup.connect(url).get();

Elements ele = doc.select(".st-ext span");

log.info(ele.text());

}

@Test //抓取商品分类下商品的数量，去除特殊链接

public void itemCatCount() throws IOException{

Integer total = 0;

String url = "http://www.jd.com/allSort.aspx";

Document doc = Jsoup.connect(url).get();

Elements level3 = doc.select("dl dd a");

log.info("三级分类总数："+level3.size());

for(Element ele : level3){

log.info(ele.text()+" "+ele.attr("href"));

String urlList = "http:"+ele.attr("href");

if(urlList.indexOf("?cat=")>0){ //有多种链接，只有含有cat才是商品列表页面

Document listDoc = Jsoup.connect(urlList).get();

Elements eleCount = listDoc.select(".st-ext span");

Integer catCount = 0;

try{

catCount = Integer.valueOf(eleCount.text());

}catch(Exception e){

catCount = 0;

}

total += catCount;

Elements elePages = listDoc.select("#J\_topPage span.fp-text i");

Integer pages = 0;

try{

pages = Integer.valueOf(elePages.text());

}catch(Exception e){

pages = 0;

}

log.info(ele.text()+" 商品数："+catCount+" 页数："+pages);

}

}

log.info("总商品数量："+total);

}

### 商品列表页面抓取商品编号

@Test //获取商品详细信息

public void getItemInfo() throws IOException{

String itemId = "1411013";

String url = "http://item.jd.com/"+itemId+".html";

Document doc = Jsoup.connect(url).get();

String title = doc.select(".sku-name").get(0).text();

log.info("标题："+title);

Elements eleImages = doc.select("div#spec-list li img");

String[] images = new String[eleImages.size()];

for(int i=0;i<eleImages.size();i++){

images[i] = eleImages.get(i).attr("src");

log.info("链接图片["+i+"]：" + images[i]);

}

}

### 抓取商品价格

京东的价格是单独发起ajax请求，返回json数组，一次可以查询多个价格

http://p.3.cn/prices/mgets?skuIds=J\_1411013,J\_1411014

返回结果为json数组：

[{"id":"J\_1411013","p":"3888.00","m":"6699.00"},{"id":"J\_1411014","p":"569.00","m":"1398.00"}]

### 抓取商品卖点

京东的卖点是单独发起ajax请求，返回json格式数据，回显到页面

http://ad.3.cn/ads/mgets?skuids=AD\_1411013,AD\_1411014

返回结果为json数组：

[{"ad":"\u53CC\u66F2\u9762\u4FA7\u5C4F\uFF0C\u91D1\u5C5E\u4E0E\u73BB\u7483\u5DE7\u5999\u878D\u5408\uFF0C\u81EA\u52A8\u8FFD\u7126\uFF0C\u667A\u80FD\u9065\u63A7\u5668\u652F\u6301\u7EA2\u5916\u9065\u63A7\u529F\u80FD\uFF01\u003C\u0061\u0020\u0020\u0074\u0061\u0072\u0067\u0065\u0074\u003D\u0022\u005F\u0062\u006C\u0061\u006E\u006B\u0022\u0020\u0020\u0068\u0072\u0065\u0066\u003D\u0022\u0068\u0074\u0074\u0070\u003A\u002F\u002F\u0073\u0061\u006C\u0065\u002E\u006A\u0064\u002E\u0063\u006F\u006D\u002F\u0061\u0063\u0074\u002F\u006F\u004C\u005A\u0052\u006C\u0057\u007A\u004D\u0070\u0049\u002E\u0068\u0074\u006D\u006C\u0022\u003E\u9886\u5238\u51CF\u94B1\uFF0C\u4E0B\u5355\u8FD4\u73B0\u003C\u002F\u0061\u003E","id":"AD\_1411013"},{"ad":"\u62C9\u6746\u7BB1\u3001\u4E66\u5305\u3001\u53CC\u80A9\u5305\u7B49\u591A\u79CD\u5546\u54C1\u9886\u5238\u6EE1\u0039\u0039\u51CF\u0032\u0030\u3001\u6EE1\u0033\u0039\u0039\u51CF\u0038\u0030\u003C\u0061\u0020\u0068\u0072\u0065\u0066\u003D\u0027\u0068\u0074\u0074\u0070\u003A\u002F\u002F\u0073\u0061\u006C\u0065\u002E\u006A\u0064\u002E\u0063\u006F\u006D\u002F\u0061\u0063\u0074\u002F\u006F\u0057\u0073\u007A\u0055\u0033\u006E\u0032\u0059\u0070\u006D\u0074\u0044\u002E\u0068\u0074\u006D\u006C\u0027\u0020\u0074\u0061\u0072\u0067\u0065\u0074\u003D\u0027\u005F\u0062\u006C\u0061\u006E\u006B\u0027\u003E\u5F00\u5B66\u4E0D\u5C06\u201C\u65E7\u201D\uFF0C\u9886\u5238\u4EAB\u6EE1\u51CF\uFF01\uFF01\u901F\u901F\u62A2\u8D2D\u5427\u007E\u003C\u002F\u0061\u003E","id":"AD\_1411014"}]

### 抓取商品描述

京东的商品描述是单独发起ajax请求，返回jsonp格式数据，回显到页面

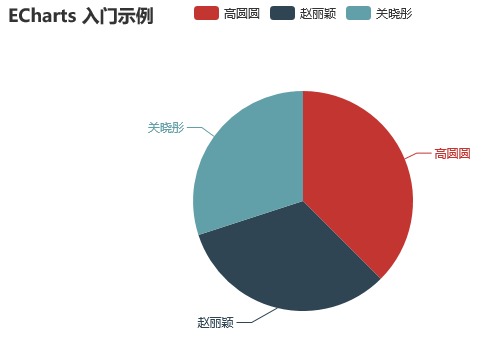
http://d.3.cn/desc/1411013

showdesc({"date":1469167322294,"content":"<div style=\"text-align: center;\"><img data-lazyload=\"//img30.360buyimg.com/jgsq-productsoa/jfs/t2767/259/3526014654/1592287/78020f87/5791889cN81b6b7f9.jpg\" alt=\"\" /><br /></div><div style=\"text-align: center;\"><img data-lazyload=\"//img30.360buyimg.com/jgsq-productsoa/jfs/t2386/186/1517820110/681125/d98ab4ac/56b1b1a3N229b52dc.jpg\" style=\"widows: auto;\" alt=\"\" /><br /></div><br/>"})

# 第三天：爬取商品价格+报表eCharts

## eCharts

### 饼形图



pie.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>ECharts</title>

<!-- 引入 echarts.js -->

<script type="text/javascript" src="js/echarts-all.js"></script>

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的dom -->

<div id="main" style="width: 600px;height:400px;"></div>

<script type="text/javascript">

// 基于准备好的dom，初始化echarts实例

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

var json = [

{value:30,name:'高圆圆'},

{value:26,name:'赵丽颖'},

{value:24,name:'关晓彤'}

];

// 指定图表的配置项和数据

var option = {

title: {

text: 'ECharts 入门示例'

},

tooltip: {},

legend: {

data:['高圆圆','赵丽颖','关晓彤']

},

series: [{

name: '女神',

type: 'pie',

radius: '55%', //饼形图半径大小

data:json

}]

};

// 使用刚指定的配置项和数据显示图表。

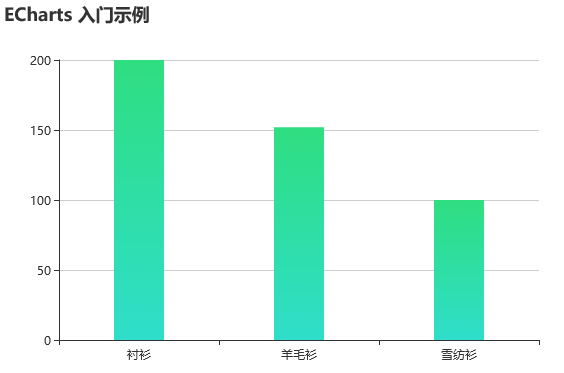
myChart.setOption(option);

</script>

</body>

</html>

### 柱状图



bar.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>ECharts</title>

<!-- 引入 echarts.js -->

<script type="text/javascript" src="js/echarts-all.js"></script>

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的dom -->

<div id="main" style="width: 600px;height:400px;"></div>

<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

var option = {

title: {

text: 'ECharts 入门示例'

},

tooltip: {},

legend: {

data:['销量']

},

xAxis: {

data: ["衬衫","羊毛衫","雪纺衫"]

},

yAxis: {},

series : [

{

name:'数量',

type:'bar',

barWidth: 50,

data:[200, 152, 100],

itemStyle: {

normal: {

color: new echarts.graphic.LinearGradient(

0, 0, 0, 1,[{offset: 0, color: '#2FDE80'},{offset: 1, color: '#2FDECA'} ]

)

},

emphasis: {

color: new echarts.graphic.LinearGradient(

0, 0, 0, 1,[{offset: 0, color: '#2FDECA'},{offset: 1, color: '#2FDE80'}]

)

}

}

}]

};

// 使用刚指定的配置项和数据显示图表。

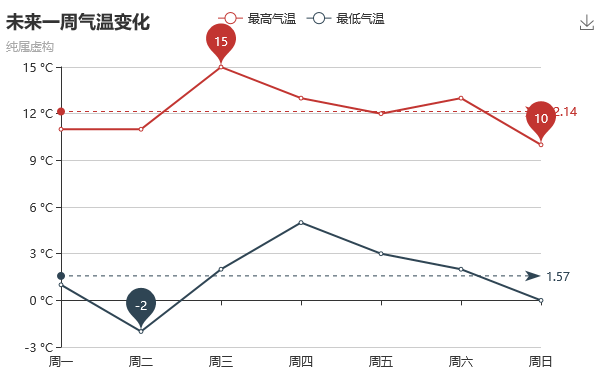
myChart.setOption(option);

</script>

</body>

</html>

### 曲线图



line.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>ECharts</title>

<!-- 引入 echarts.js -->

<script type="text/javascript" src="js/echarts-all.js"></script>

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的dom -->

<div id="main" style="height:400px;width: 600px"></div>

<!-- ECharts单文件引入 -->

<script src="http://echarts.baidu.com/build/dist/echarts-all.js"></script>

<script type="text/javascript">

// 基于准备好的dom，初始化echarts图表

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

var option = {

title : {

text: '未来一周气温变化',

subtext: '纯属虚构'

},

tooltip : {

trigger: 'axis'

},

legend: {

data:['最高气温','最低气温']

},

//右上角工具条

toolbox: {

show : true,

feature : {

saveAsImage : {show: true}

}

},

calculable : true,

xAxis : [

{

type : 'category',

boundaryGap : false,

data : ['周一','周二','周三','周四','周五','周六','周日']

}

],

yAxis : [

{

type : 'value',

axisLabel : {

formatter: '{value} °C'

}

}

],

series : [

{

name:'最高气温',

type:'line',

data:[11, 11, 15, 13, 12, 13, 10],

markPoint : {

data : [

{type : 'max', name: '最大值'},

{type : 'min', name: '最小值'}

]

},

markLine : {

data : [

{type : 'average', name: '平均值'}

]

}

},

{

name:'最低气温',

type:'line',

data:[1, -2, 2, 5, 3, 2, 0],

markPoint : {

data : [

{type : 'min', name: '周最低'}

]

},

markLine : {

data : [

{type : 'average', name : '平均值'}

]

}

}

]

};

// 为echarts对象加载数据

myChart.setOption(option);

</script>

</body>

</html>

## 比价器系统

### 京东

京东的价格是单独发起ajax请求，返回json数组，一次可以查询多个价格

http://p.3.cn/prices/mgets?skuIds=J\_1411013,J\_1411014

返回结果为json数组：

[{"id":"J\_1411013","p":"3888.00","m":"6699.00"},{"id":"J\_1411014","p":"569.00","m":"1398.00"}]

### 淘宝

@Test

public void taobaoItemPrice() throws IOException{

String itemId = "560563554738";

String url = "http://mdskip.taobao.com/core/initItemDetail.htm?isRegionLevel=true&itemTags=385,775,843,1035,1163,1227,1478,1483,1539,1611,1863,1867,1923,2049,2059,2242,2251,2315,2507,2635,3595,3974,4166,4299,4555,4811,5259,5323,5515,6145,6785,7809,9153,11265,12353,12609,13697,13953,16321,16513,17473,17537,17665,17857,18945,19841,20289,21762,21826,25922,28802,53954&tgTag=false&addressLevel=4&isAreaSell=false&sellerPreview=false&offlineShop=false&showShopProm=false&isIFC=false&service3C=true&isSecKill=false&isForbidBuyItem=false&cartEnable=true&sellerUserTag=839979040&queryMemberRight=true&itemId="+itemId+"&sellerUserTag2=306250462070310924&household=false&isApparel=false¬AllowOriginPrice=false&tmallBuySupport=true&sellerUserTag3=144467169269284992&sellerUserTag4=1152930305168967075&progressiveSupport=true&isUseInventoryCenter=false&tryBeforeBuy=false&callback=setMdskip×tamp=1420351892310";

HttpClientBuilder builder = HttpClients.custom();

builder.setUserAgent("Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:0.9.4)");

CloseableHttpClient httpClient = builder.build();

final HttpGet httpGet = new HttpGet(url);

httpGet.addHeader("Referer", "http://detail.tmall.com/item.htm?id="+itemId+"&skuId=68347779144&areaId=110000&cat\_id=50024400&rn=763d147479ecdc17c2632a4219ce96b3&standard=1&user\_id=263726286&is\_b=1");

CloseableHttpResponse response = null;

response = httpClient.execute(httpGet);

final HttpEntity entity = response.getEntity();

String result = null;

if (entity != null) {

result = EntityUtils.toString(entity);

EntityUtils.consume(entity);

}

//商品价格的返回值，需要解析出来价格

result = result.substring(10, result.length()-1);

}

### 苏宁

http://ds.suning.cn/ds/generalForTile/000000000690128134-9173-2-0000000000-1--ds000000000.jsonp

### 价格比较柱状图

price.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>ECharts</title>

<!-- 引入 echarts.js -->

<script type="text/javascript" src="js/echarts-all.js"></script>

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的dom -->

<div id="main" style="width: 600px;height:400px;"></div>

<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

var option = {

title: {

text: '各大电商商品价格比价'

},

tooltip: {},

legend: {

data:['iphoneX 64g']

},

xAxis: {

data: ["京东","淘宝","苏宁"]

},

yAxis: {},

series : [

{

name:'数量',

type:'bar',

barWidth: 50,

data:[8316, 6783, 7788],

itemStyle: {

normal: {

color: new echarts.graphic.LinearGradient(

0, 0, 0, 1,[{offset: 0, color: '#2FDE80'},{offset: 1, color: '#2FDECA'} ]

)

},

emphasis: {

color: new echarts.graphic.LinearGradient(

0, 0, 0, 1,[{offset: 0, color: '#2FDECA'},{offset: 1, color: '#2FDE80'}]

)

}

}

}]

};

// 使用刚指定的配置项和数据显示图表。

myChart.setOption(option);

</script>

</body>

</html>