

BusinessViewService WebService 示例.

下面以Java环境为例，介绍如何在应用系统中调用Smartbi WebService BusinessViewService API。

1、通过wsimport生成Java的WebService客户端实现

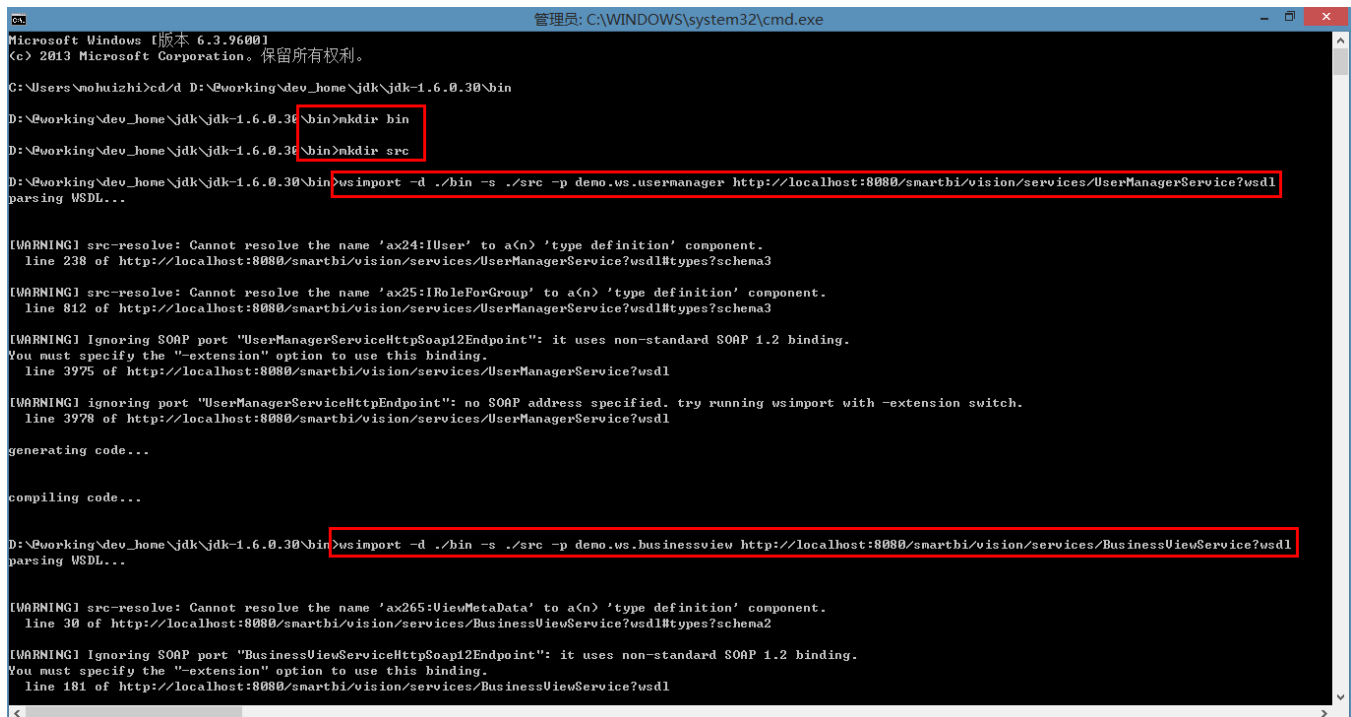
wsimport是在JDK(1.6或以上)的bin目录下的一个exe文件（Windows版），主要功能是根据服务端发布的wsdl文件生成客户端存根及框架，负责与Web Service 服务器通信，并在将其封装成实例，客户端可以直接使用，就像使用本地实例一样。对Java而言，wsimport帮助程序员生存调用web service所需要的客户端类文件.java和.class。要提醒指出的是，wsimport可以用于非Java的服务器端，如：服务器端也许是C# 编写的web service。

wsimport命令的参数说明：

- -d 生成客户端执行类的class文件的存放目录
 - -s 生成客户端执行类的源文件的存放目录
 - -p 定义生成类的包名
- 更多参数说明请执行：wsimport -help

打开Windows命令行程序，通过以下命令生成UserManagerService与BusinessViewService的WebService客户端实现源码(在本示例文档中UserManagerService主要用于登录)：

- wsimport -d ./bin -s ./src -p demo.ws.usermanager <http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl>
 - wsimport -d ./bin -s ./src -p demo.ws.businessview <http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl>
- 命令执行效果如下：



```
管理员: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation. 保留所有权利。

C:\Users\mohuizhi>cd /d D:\Working\dev_home\jdk\jdk-1.6.0.30\bin

D:\Working\dev_home\jdk\jdk-1.6.0.30>bin>mkdir bin
D:\Working\dev_home\jdk\jdk-1.6.0.30>bin>mkdir src

D:\Working\dev_home\jdk\jdk-1.6.0.30>bin>wsimport -d ./bin -s ./src -p demo.ws.usermanager http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl
parsing WSDL...

[WARNING] src-resolve: Cannot resolve the name 'ax24:IUser' to a(n) 'type definition' component.
  line 238 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl#types?schema3

[WARNING] src-resolve: Cannot resolve the name 'ax25:IRoleForGroup' to a(n) 'type definition' component.
  line 812 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl#types?schema3

[WARNING] Ignoring SOAP port "UserManagerServiceHttpSoap12Endpoint": it uses non-standard SOAP 1.2 binding.
  You must specify the "-extension" option to use this binding.
  line 3975 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl

[WARNING] ignoring port "UserManagerServiceHttpEndpoint": no SOAP address specified. try running wsimport with -extension switch.
  line 3978 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl

generating code...

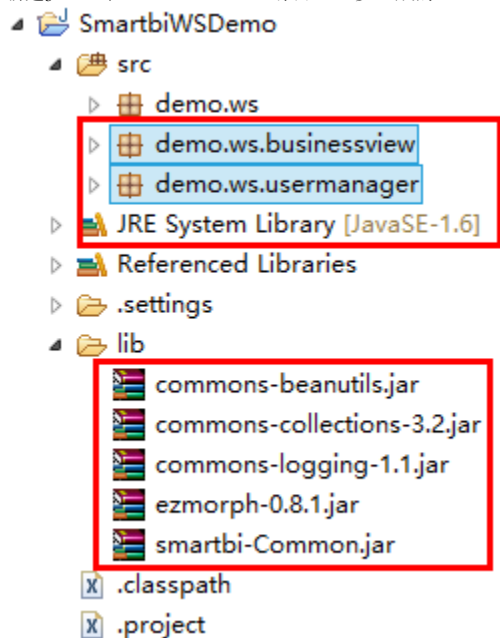
compiling code...

D:\Working\dev_home\jdk\jdk-1.6.0.30>bin>wsimport -d ./bin -s ./src -p demo.ws.businessview http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl
parsing WSDL...

[WARNING] src-resolve: Cannot resolve the name 'ax265:ViewMetaData' to a(n) 'type definition' component.
  line 30 of http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl#types?schema2

[WARNING] Ignoring SOAP port "BusinessViewServiceHttpSoap12Endpoint": it uses non-standard SOAP 1.2 binding.
  You must specify the "-extension" option to use this binding.
  line 181 of http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl
```

2、新建Java工程SmartbiWSDemo，添加上一步生成的WebService客户端实现的Java文件以及所依赖的jar：



3、核心示例代码如下：

```
package demo.ws;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.xml.ws.BindingProvider;
import smartbi.net.sf.json.JSONArray;
import smartbi.net.sf.json.JSONObject;
import demo.ws.businessview.ArrayOfArrayOfString;
import demo.ws.businessview.ArrayOfString;
import demo.ws.businessview.BusinessViewService;
import demo.ws.businessview.BusinessViewServicePortType;
import demo.ws.businessview.ViewMetaData;
import demo.ws.usermanager.UserManagerService;
import demo.ws.usermanager.UserManagerServicePortType;

public class BusinessViewServiceWSDemo {
    private static final String KEY_RESPONSE_HEADERS = "javax.xml.ws.http.response.headers";
    private static final String KEY_REQUEST_HEADERS = "javax.xml.ws.http.request.headers";
    private static List<String> cookieHeaders = null;
    /**
     * @param args
     */
    public static void main(String[] args) {
        BusinessViewServiceWSDemo demo = new BusinessViewServiceWSDemo();
        demo.run();
    }
    /**
     * BusinessViewService WebService
     */
    private void run() {
        if (!login()) {
            System.out.println("url");
            return;
        }
        try {
            testBusinessViewService();
        } finally {
        }
    }
}
```

```

        logout();
    }
}
/**
 *
 *
 * @return
 */
@SuppressWarnings("unchecked")
private boolean login() {
    String user = "admin";
    String password = "manager";
    //
    UserManagerServicePortType userServicePort = new UserManagerService()
        .getUserManagerServiceHttpSoap11Endpoint();
    BindingProvider bp = (BindingProvider) userServicePort;
    // bp.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
    boolean success = userServicePort.login(user, password);
    if (success) {
        Map<String, List<String>> headers = (Map<String, List<String>>) bp.
getResponseContext()
        .get(KEY_RESPONSE_HEADERS);
        // Retrieve the cookie from the response message
        cookieHeaders = headers.get("Set-Cookie");
    }
    return success;
}
/**
 * uses HTTP cookies to establish credentials with the server
 */
@SuppressWarnings("unchecked")
private void retrieveCookie(Map<String, Object> requestContext) {
    Map<String, List<String>> headers = (Map<String, List<String>>) requestContext
        .get(KEY_REQUEST_HEADERS);
    if (headers == null) {
        headers = new HashMap<String, List<String>>();
        requestContext.put(KEY_REQUEST_HEADERS, headers);
    }
    headers.put("cookie", cookieHeaders);
}
/**
 *
 *
 */
private void testBusinessViewService() {
    BusinessViewServicePortType bizViewServicePort = new BusinessViewService()
        .getBusinessViewServiceHttpSoap11Endpoint();
    BindingProvider bp = (BindingProvider) bizViewServicePort;
    // bp.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
    retrieveCookie(bp.getRequestContext());
    // ID("\\SQL\\SQL")
    String bizViewId = "I2c94907127855da201278df551b62602";
    // ID
    String paramId = "OutputParameter.I2c94907127855da201278df551b62602._";
    //
    String paramValue = ",,,";
    //
    // String paramDisplayValue = paramValue;
    //
    JSONArray paramsArr = new JSONArray();
    JSONObject paramObj = new JSONObject();
    paramObj.put("id", paramId);
    paramObj.put("value", paramValue);
    // paramObj.put("displayValue", paramDisplayValue);
    paramsArr.put(paramObj);
    String paramsJsonArrStr = paramsArr.toString();
    //
    //
    int rowsPerPage = 2;
    //
    boolean getTotalRows = true;
    //

```

```

ViewMetaData ret = bizViewServicePort.openLoadDataView(bizViewId, paramsJsonArrStr,
    rowsPerPage, getTotalRows);
String loadDataClientId = ret.getClientId();
int totalRows = ret.getTotalRowCount();
List<String> fieldNames = ret.getFieldNames();
// List<String> fieldTypes = ret.getFieldTypes();
//
int pages = totalRows / rowsPerPage + (totalRows % rowsPerPage == 0 ? 0 : 1);
System.out.println("" + totalRows + "" + rowsPerPage + "" + pages + "");
StringBuilder buf = new StringBuilder();
buf.append("|\\t");
for (String name : fieldNames) {
    buf.append(name).append("\\t");
}
buf.append("|");
System.out.println("-----");
System.out.println(buf.toString());
//
for (int pageNum = 0; pageNum < pages; pageNum++) {
    System.out.println("---" + (pageNum + 1) + "-----");
    ArrayOfArrayOfString dataList = bizViewServicePort.loadViewData(loadDataClientId,
        pageNum);
    if (dataList != null && dataList.getArray().size() > 0) {
        List<ArrayOfString> cellDatas = dataList.getArray();
        for (ArrayOfString cellDataArr : cellDatas) {
            buf.setLength(0);
            buf.append("|\\t");
            for (String data : cellDataArr.getArray()) {
                buf.append(data).append("\\t");
            }
            buf.append("|");
            System.out.println(buf.toString());
        }
    }
}
System.out.println("-----");
//
bizViewServicePort.closeLoadDataView(loadDataClientId);
}
/**
 *
 */
private boolean logout() {
    UserManagerServicePortType userServicePort = new UserManagerService()
        .getUserManagerServiceHttpSoap11Endpoint();
    BindingProvider bp = (BindingProvider) userServicePort;
    retrieveCookie(bp.getRequestContext());
    return userServicePort.logout();
}
}

```

4、示例运行效果:

<terminated> BusinessViewServiceWSDemo (1) [Java Application] D:\@working\dev_home\jdk\jdk-1.6.0.30\bin\javaw.exe (
共5行，每页2行，共3页：				
行号	产品名称	单价	库存	
---第1页---				
1	牛奶	19.0	17.0	
2	饼干	17.45	29.0	
---第2页---				
3	猪肉	39.0	0.0	
4	花生	10.0	3.0	
---第3页---				
5	汽水	4.5	20.0	

5、示例源码(Eclipse)工程

[SmartbiWSDemo_Eclipse_Project.rar](#)

下面以Java环境为例，介绍如何在应用系统中调用Smartbi WebService BusinessViewService API。

1、通过wsimport生成Java的WebService客户端实现

wsimport是在JDK(1.6或以上)的bin目录下的一个exe文件（Windows版），主要功能是根据服务端发布的wsdl文件生成客户端存根及框架，负责与Web Service 服务器通信，并在将其封装成实例，客户端可以直接使用，就像使用本地实例一样。对Java而言，wsimport帮助程序员生存调用web service所需要的客户端类文件.java和.class。要提醒指出的是，wsimport可以用于非Java的服务器端，如：服务器端也许是C# 编写的web service。

wsimport命令的参数说明：

- -d 生成客户端执行类的class文件的存放目录
 - -s 生成客户端执行类的源文件的存放目录
 - -p 定义生成类的包名
- 更多参数说明请执行：wsimport -help

打开Windows命令行程序，通过以下命令生成UserManagerService与BusinessViewService的WebService客户端实现源码(在本示例文档中UserManagerService主要用于登录)：

- wsimport -d ./bin -s ./src -p demo.ws.usermanager http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl
 - wsimport -d ./bin -s ./src -p demo.ws.businessview http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl
- 命令执行效果如下：

```
管理员: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Users\mohuizhi>cd/d D:\Working\dev_home\jdk\jdk-1.6.0.30\bin
D:\Working\dev_home\jdk\jdk-1.6.0.30\bin>mkdir bin
D:\Working\dev_home\jdk\jdk-1.6.0.30\bin>mkdir src
D:\Working\dev_home\jdk\jdk-1.6.0.30\bin>wsimport -d ./bin -s ./src -p demo.ws.usermanager http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl
parsing WSDL...

[WARNING] src-resolve: Cannot resolve the name 'ax24:IUser' to a(n) 'type definition' component.
  line 238 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl#types?schema3
[WARNING] src-resolve: Cannot resolve the name 'ax25:IRoleForGroup' to a(n) 'type definition' component.
  line 812 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl#types?schema3
[WARNING] Ignoring SOAP port "UserManagerServiceHttpSoap12Endpoint": it uses non-standard SOAP 1.2 binding.
You must specify the "-extension" option to use this binding.
  line 3975 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl
[WARNING] ignoring port "UserManagerServiceHttpEndpoint": no SOAP address specified. try running wsimport with -extension switch.
  line 3978 of http://localhost:8080/smartbi/vision/services/UserManagerService?wsdl

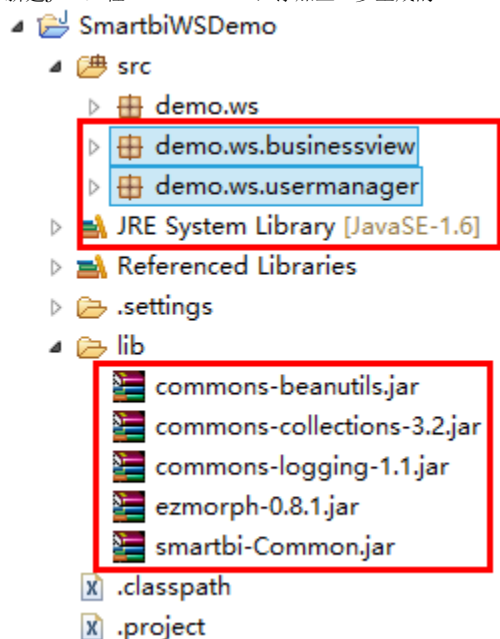
generating code...

compiling code...

D:\Working\dev_home\jdk\jdk-1.6.0.30\bin>wsimport -d ./bin -s ./src -p demo.ws.businessview http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl
parsing WSDL...

[WARNING] src-resolve: Cannot resolve the name 'ax265:ViewMetaData' to a(n) 'type definition' component.
  line 30 of http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl#types?schema2
[WARNING] Ignoring SOAP port "BusinessViewServiceHttpSoap12Endpoint": it uses non-standard SOAP 1.2 binding.
You must specify the "-extension" option to use this binding.
  line 181 of http://localhost:8080/smartbi/vision/services/BusinessViewService?wsdl
```

2、新建Java工程SmartbiWSDemo，添加上一步生成的WebService客户端实现的Java文件以及所依赖的jar：



3、核心示例代码如下：

```
package demo.ws;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.xml.ws.BindingProvider;
import smartbi.net.sf.json.JSONArray;
import smartbi.net.sf.json.JSONObject;
import demo.ws.businessview.ArrayOfArrayOfString;
```

```

import demo.ws.businessview.ArrayOfString;
import demo.ws.businessview.BusinessViewService;
import demo.ws.businessview.BusinessViewServicePortType;
import demo.ws.businessview.ViewMetaData;
import demo.ws.usermanager.UserManagerService;
import demo.ws.usermanager.UserManagerServicePortType;

public class BusinessViewServiceWSDemo {
    private static final String KEY_RESPONSE_HEADERS = "javax.xml.ws.http.response.headers";
    private static final String KEY_REQUEST_HEADERS = "javax.xml.ws.http.request.headers";
    private static List<String> cookieHeaders = null;
    /**
     * @param args
     */
    public static void main(String[] args) {
        BusinessViewServiceWSDemo demo = new BusinessViewServiceWSDemo();
        demo.run();
    }
    /**
     * BusinessViewService WebService
     */
    private void run() {
        if (!login()) {
            System.out.println("url");
            return;
        }
        try {
            testBusinessViewService();
        } finally {
            logout();
        }
    }
    /**
     *
     *
     * @return
     */
    @SuppressWarnings("unchecked")
    private boolean login() {
        String user = "admin";
        String password = "manager";
        //
        UserManagerServicePortType userServicePort = new UserManagerService()
            .getUserManagerServiceHttpSoap11Endpoint();
        BindingProvider bp = (BindingProvider) userServicePort;
        // bp.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
        boolean success = userServicePort.login(user, password);
        if (success) {
            Map<String, List<String>> headers = (Map<String, List<String>>) bp.
getResponseContext()
                .get(KEY_RESPONSE_HEADERS);
            // Retrieve the cookie from the response message
            cookieHeaders = headers.get("Set-Cookie");
        }
        return success;
    }
    /**
     * uses HTTP cookies to establish credentials with the server
     */
    @SuppressWarnings("unchecked")
    private void retrieveCookie(Map<String, Object> requestContext) {
        Map<String, List<String>> headers = (Map<String, List<String>>) requestContext
            .get(KEY_REQUEST_HEADERS);
        if (headers == null) {
            headers = new HashMap<String, List<String>>();
            requestContext.put(KEY_REQUEST_HEADERS, headers);
        }
        headers.put("cookie", cookieHeaders);
    }
    /**

```

```

*
*/
private void testBusinessViewService() {
    BusinessViewServicePortType bizViewServicePort = new BusinessViewService()
        .getBusinessViewServiceHttpSoap11Endpoint();
    BindingProvider bp = (BindingProvider) bizViewServicePort;
    // bp.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
    retrieveCookie(bp.getRequestContext());
    // ID("\\SQL\\SQL")
    String bizViewId = "I2c94907127855da201278df551b62602";
    // ID
    String paramId = "OutputParameter.I2c94907127855da201278df551b62602._";
    //
    String paramValue = ",,,";
    //
    // String paramDisplayValue = paramValue;
    //
    JSONArray paramsArr = new JSONArray();
    JSONObject paramObj = new JSONObject();
    paramObj.put("id", paramId);
    paramObj.put("value", paramValue);
    // paramObj.put("displayValue", paramDisplayValue);
    paramsArr.put(paramObj);
    String paramsJsonArrStr = paramsArr.toString();
    //
    //
    int rowsPerPage = 2;
    //
    boolean getTotalRows = true;
    //
    ViewMetaData ret = bizViewServicePort.openLoadDataView(bizViewId, paramsJsonArrStr,
        rowsPerPage, getTotalRows);
    String loadDataClientId = ret.getClientId();
    int totalRows = ret.getTotalRowCount();
    List<String> fieldNames = ret.getFieldNames();
    // List<String> fieldTypes = ret.getFieldTypes();
    //
    int pages = totalRows / rowsPerPage + (totalRows % rowsPerPage == 0 ? 0 : 1);
    System.out.println(" " + totalRows + " " + rowsPerPage + " " + pages + " ");
    StringBuilder buf = new StringBuilder();
    buf.append(" | \t ");
    for (String name : fieldNames) {
        buf.append(name).append(" \t ");
    }
    buf.append(" | ");
    System.out.println("-----");
    System.out.println(buf.toString());
    //
    for (int pageNum = 0; pageNum < pages; pageNum++) {
        System.out.println("---" + (pageNum + 1) + "-----");
        ArrayOfArrayOfString dataList = bizViewServicePort.loadViewData(loadDataClientId,
            pageNum);
        if (dataList != null && dataList.getArray().size() > 0) {
            List<ArrayOfString> cellDatas = dataList.getArray();
            for (ArrayOfString cellDataArr : cellDatas) {
                buf.setLength(0);
                buf.append(" | \t ");
                for (String data : cellDataArr.getArray()) {
                    buf.append(data).append(" \t ");
                }
                buf.append(" | ");
                System.out.println(buf.toString());
            }
        }
    }
    System.out.println("-----");
    //
    bizViewServicePort.closeLoadDataView(loadDataClientId);
}
/**
*

```



```

        */
private boolean logout() {
    UserManagerServicePortType userServicePort = new UserManagerService()
        .getUserManagerServiceHttpSoap11Endpoint();
    BindingProvider bp = (BindingProvider) userServicePort;
    retrieveCookie(bp.getRequestContext());
    return userServicePort.logout();
}
}

```

4、示例运行效果：

Console Problems History Search Progress

<terminated> BusinessViewServiceWSDemo (1) [Java Application] D:\@working\dev_home\jdk\jdk-1.6.0.30\bin\javaw.exe (

共5行，每页2行，共3页：

行号	产品名称	单价	库存
---第1页---			
1	牛奶	19.0	17.0
2	饼干	17.45	29.0
---第2页---			
3	猪肉	39.0	0.0
4	花生	10.0	3.0
---第3页---			
5	汽水	4.5	20.0