

Developing JSF 2.0 Portlet Applications using Primefaces with Rational Application Developer 8.5

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Table of Contents

Abstract.....	3
Overview.....	3
Sample application	3
Developing the sample application	4
Creating a Faces Library Definition.....	4
Portlet Project and Portlet Creation.....	7
Developing Market Watch Portlet.....	11
Creating scoped managed bean.....	11
Custom code specific to the application.....	16
Developing GUI using Primefaces widgets.....	16
Developing Trading Portlet.....	19
Creating scoped managed bean.....	22
Custom code specific to the application.....	23
Custom code specific to the application.....	24
Developing GUI using Primefaces widgets.....	25
Developing Portfolio Portlet.....	33
Creating scoped managed bean.....	33
Custom code specific to the application.....	34
Custom code specific to the application.....	34
Developing GUI using Primefaces widgets.....	35
Developing Watchlist portlet.....	37
Creating scoped managed bean.....	37
Custom code specific to the application.....	38
Custom code specific to the application.....	38
Developing GUI using Primefaces widgets.....	38
Deploying application on WebSphere Portal server.....	41
Executing the application on the server.....	43
Market Watch Portlet.....	43
Trading Portlet	44
Portfolio Portlet.....	46
Watchlist Portlet.....	46
Buy and Sell Operations.....	47
Buy Operation from Trading Portlet.....	47
Sell Operation from Trading Portlet.....	51
Watching operation from Trading portlet	55
Removing transactions from Watchlist portlet.....	56
Conclusion.....	58
References.....	58

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Abstract

The article illustrates procedure to create and publish a JSF 2.0 portlet project on IBM WebSphere Portal v8.0 server using IBM Rational Application Developer. A sample Market Snapshot application, showcases the usage of JSF 2.0 features in a JSR 286 portlet application. Detailed instructions in the article shows some of the key JSF 2.0 features, such as Annotations, Ajax, and Resource Loading. The sample application that is developed in the article can be deployed on IBM® WebSphere® Portal version 8.0.

Overview

Java Server Faces(JSF) is a popular technology to build web applications these days. After combining the capabilities of JSF with IBM WebSphere Portal Server, then you get a combination which is hard to beat in the industry. Not only that, you still have the choice to customize your web application behavior by using features provided by a third party component libraries of JSF. This article discusses one such library – Primefaces. It demonstrates how you can use an ensemble of these technologies to build a powerful and compelling web applications.

Prerequisites

1. IBM® Rational® Application Developer version 8.5
2. IBM WebSphere Portal Server version 8.0
3. Primefaces version 3.0 component Java archive

Sample application

The Stock Market sample application is a JSF v2.0 based JSR 286 portlet application for a financial domain. The application comprises of four portlets:

- Market Watch – This portlet displays latest market trends from different stock exchanges. It also shows intra-day graphs for the exchanges.
- Trading – This portlet searches, buys, sells and adds stocks to the watchlist from different exchanges. Go to help mode of Trading portlet to check the list of stocks in the database.
- Portfolio – This portlet displays portfolio stocks in an account. You can buy or sell the stocks from Portfolio portlet as well.
- Watchlist – This portlet displays list of stocks that are on the watch list of an account. You can buy, sell, or remove the stocks from the watch list.

This sample application leverages the JSF 2.0 features and demonstrates how the following technology features are consumed in the portlet application using intuitive tools available with IBM® Rational® Application Developer version 8.5

- Ajax
- Annotations
- Composite components
- Implicit Navigation
- Resource loading
- Template

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

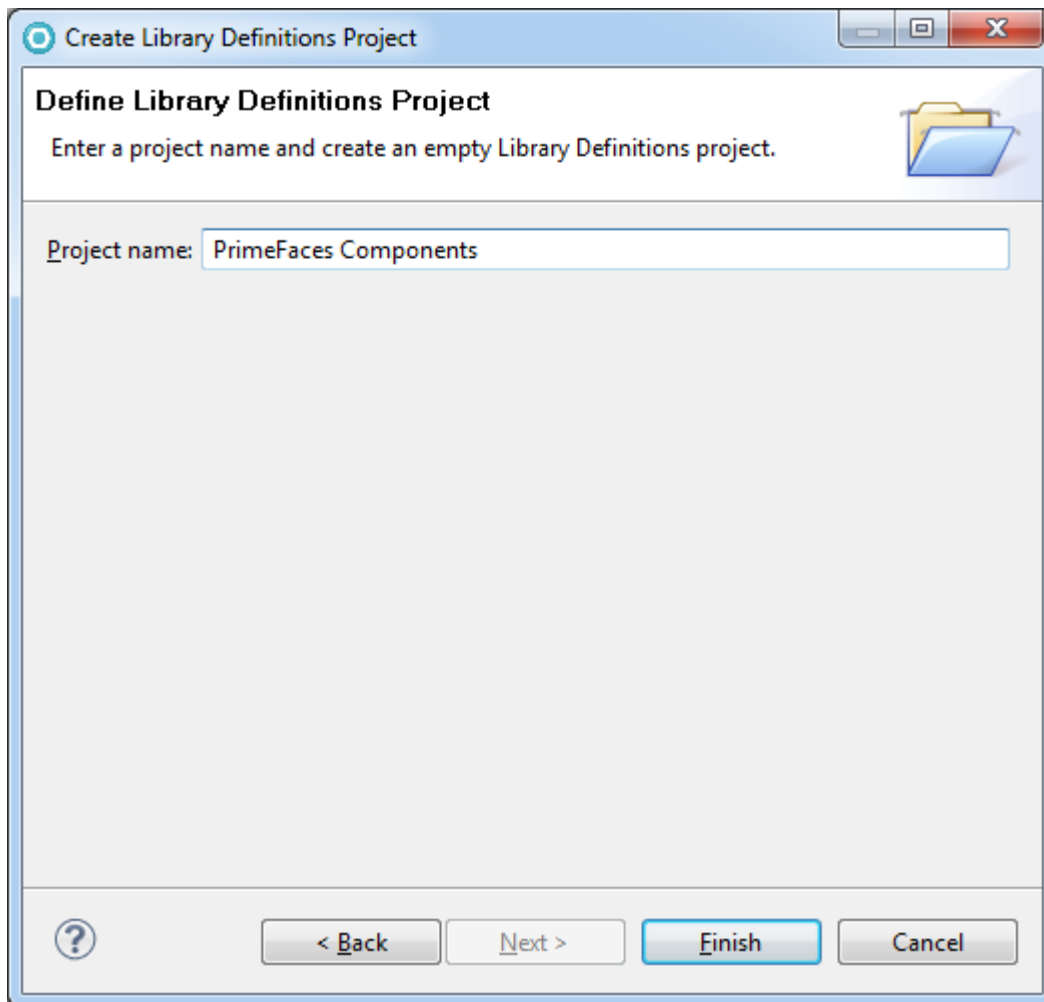
Developing the sample application

Creating a Faces Library Definition

To develop the sample application, first download the PrimeFaces component Java™ archive (JAR) file from the PrimeFaces Web site (<http://primefaces.org/downloads.html>), which is used as a third-party component library. Start by creating a Library Definitions Project to contain the library's configuration metadata.

1. Select **File > New > Project > Web > Library Definitions Project** to display the Library Definitions Project wizard.
2. Enter the project a name (in this case, **PrimeFaces Components**), as shown in Figure 1. Click **Finish**.

Figure 1. Create a Faces Definitions Project

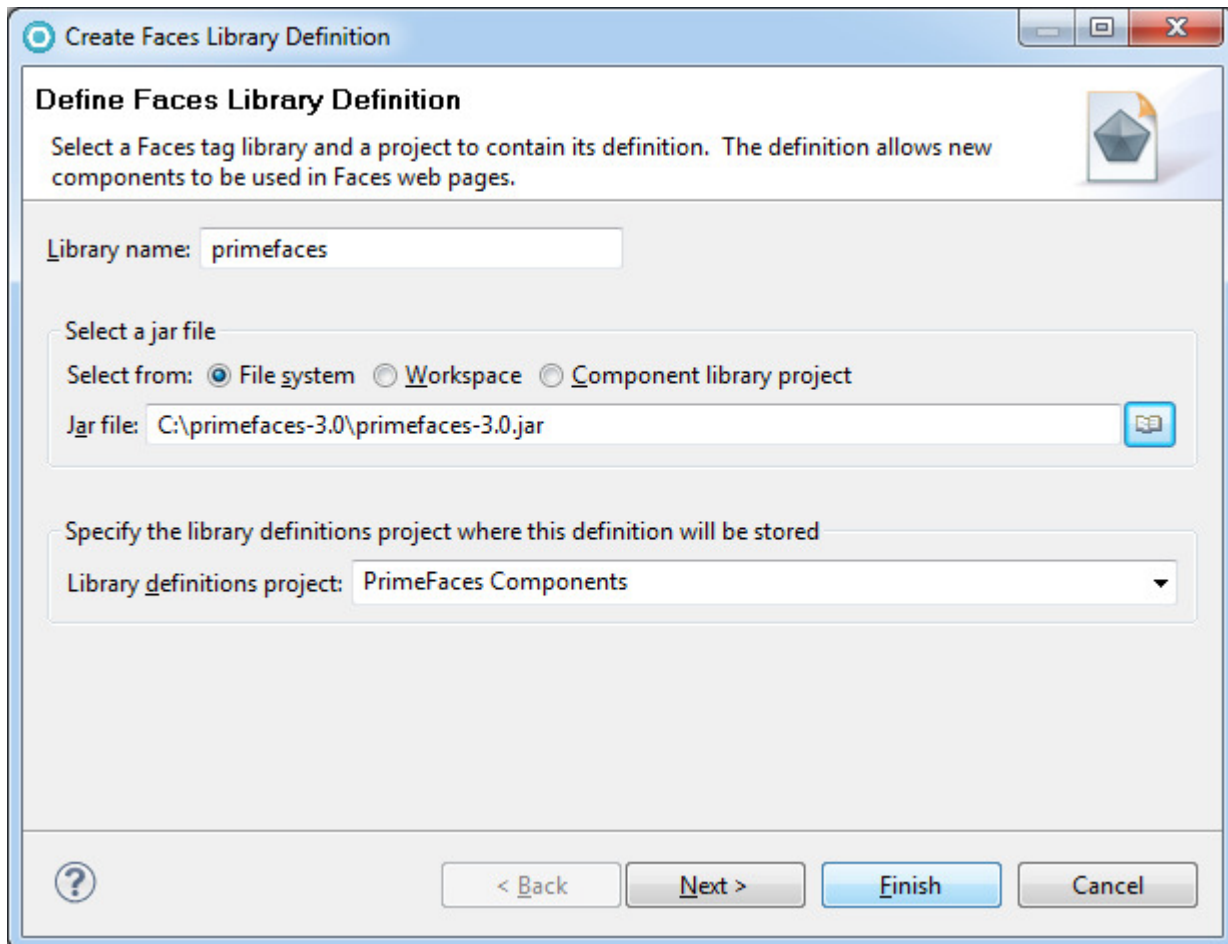


The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

To create a library definition for the PrimeFaces library and examine its elements: Right-click the new project that you just created and select **New > Faces Library Definition**. Enter the new definition a name (for example, primefaces).

1. Click the browse button next to the **Jar file** field. Select the location of the JAR file that you downloaded from the Primefaces site, as shown in Figure 2. Click **Finish**.

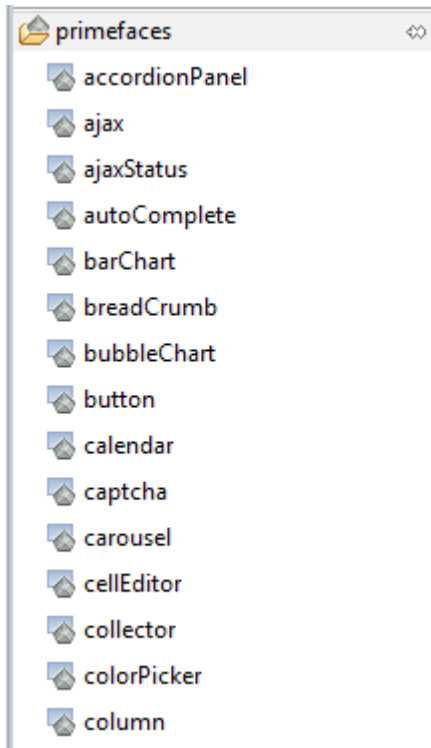
Figure 2. Create a new Faces Library Definition



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. After a short time spent analyzing the PrimeFaces JAR file and generating files, the wizard completes and opens the new JLD configuration file (primefaces.jld) in the editor.
3. When the portlet project created and facelet page opens in the editor, the new Primefaces drawer in the palette view, as shown in Figure 3.

Figure 3: Palette with primefaces components



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Portlet Project and Portlet Creation

Now create a JSF 2.0 portlet project to see how the new component library integrates with the JSF tools.

1. Select **File > New > Project > Portlet Project** to display the project wizard.
2. Enter **StockSample_Primefaces** as the portlet project name.
3. In the **Portlet settings** group, **Create a portlet** check box is selected by default. Specify the portlet name as **Market Watch**.
4. Select **WebSphere Portal v8.0** as the **Target Runtime**, as shown in Figure 4.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Figure 4: Portlet Project Creation Wizard

Portlet Project
Specify a name and location for the new portlet project.

Project name:

Project location
 Use default location
Location:

Target runtime

EAR membership
 Add project to an EAR
EAR project name:

Portlet settings
 Create a portlet
Portlet name:

Configuration: JSR286 Faces

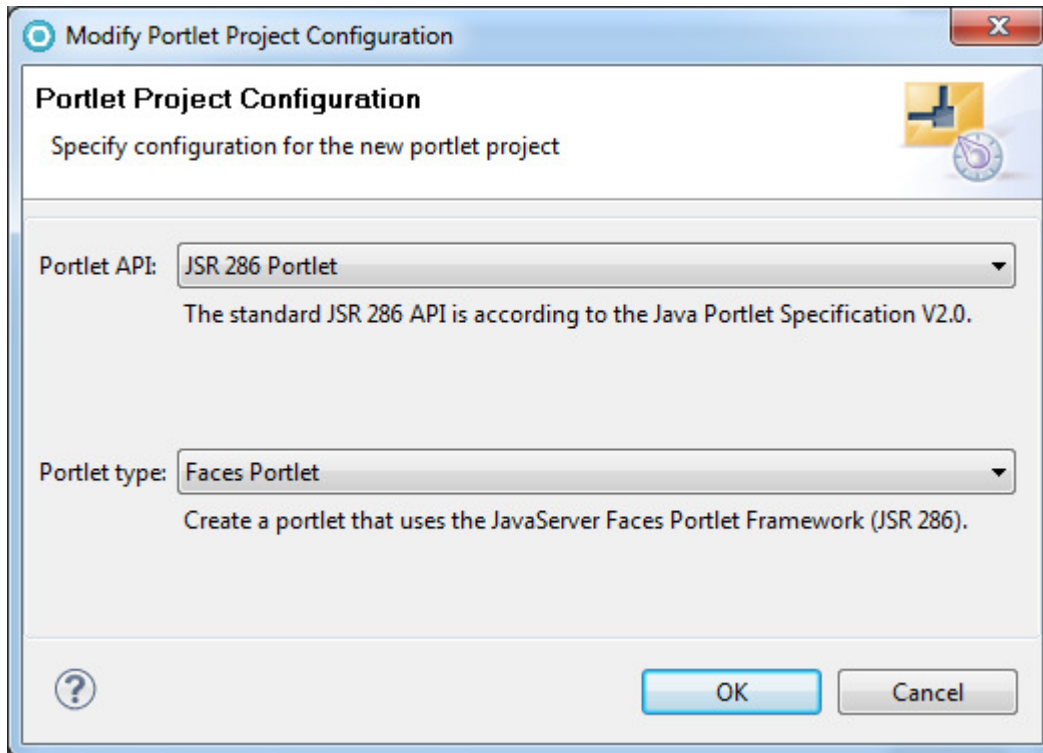
Web 2.0 Features: Client-side capabilities

Devices supported: None

5. Click **Modify** button beside Configuration to display the **Portlet Project Configuration** wizard.
6. Select **JSR 286** API and **Faces** as the Portlet Type, as shown in Figure 5. Click **OK**.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Figure 5: Select Portlet API and type



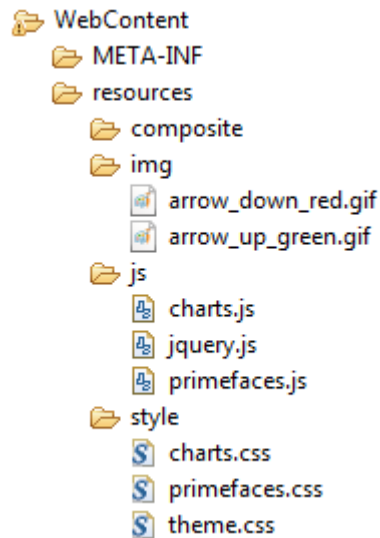
7. Click **Finish**. When prompted, switch to the Web perspective. A portlet project and the corresponding EAR are created, and it appears in the Enterprise Explorer view.
8. Given that the sample application covers four portlets. The next step is to add rest of the portlets to the portlet application. To add portlets to the portlet project:
 - Right-click the portlet deployment descriptor node and select **New > Portlet**.
 - Enter the portlet name as **Trading**, and click **Finish**.
 - Repeat these steps to add the other two portlets: **Portfolio** and **Watchlist**.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

9. Create following folder structure and copy all images, JavaScript and CSS files (copy the files from the **WebContent\resources** folder in the sample attached with this article) to the folders respectively, as shown in Figure 7:

- WebContent\resources\img
- WebContent\resources\js
- WebContent\resources\style

Figure 7: Location of image, js and css files



In this application, you will use Extensible Markup Language (XML), which is in lightweight interchange format, and presents data which is easy to parse and is in human readable form. The data provided by the XML file is copied to the data table. Copy the following XML files (from the **WebContent** folder in the attached sample) to the **WebContent** folder of the project.

- exchange.xml
- trading.xml
- portfolio.xml
- watchlist.xml

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Developing Market Watch Portlet

This portlet shows the trends for different exchanges and has an intra-day graph. It also shows the details of the various exchanges in a tabular form. Thus, creates a managed bean with the properties for the columns shown in the table.

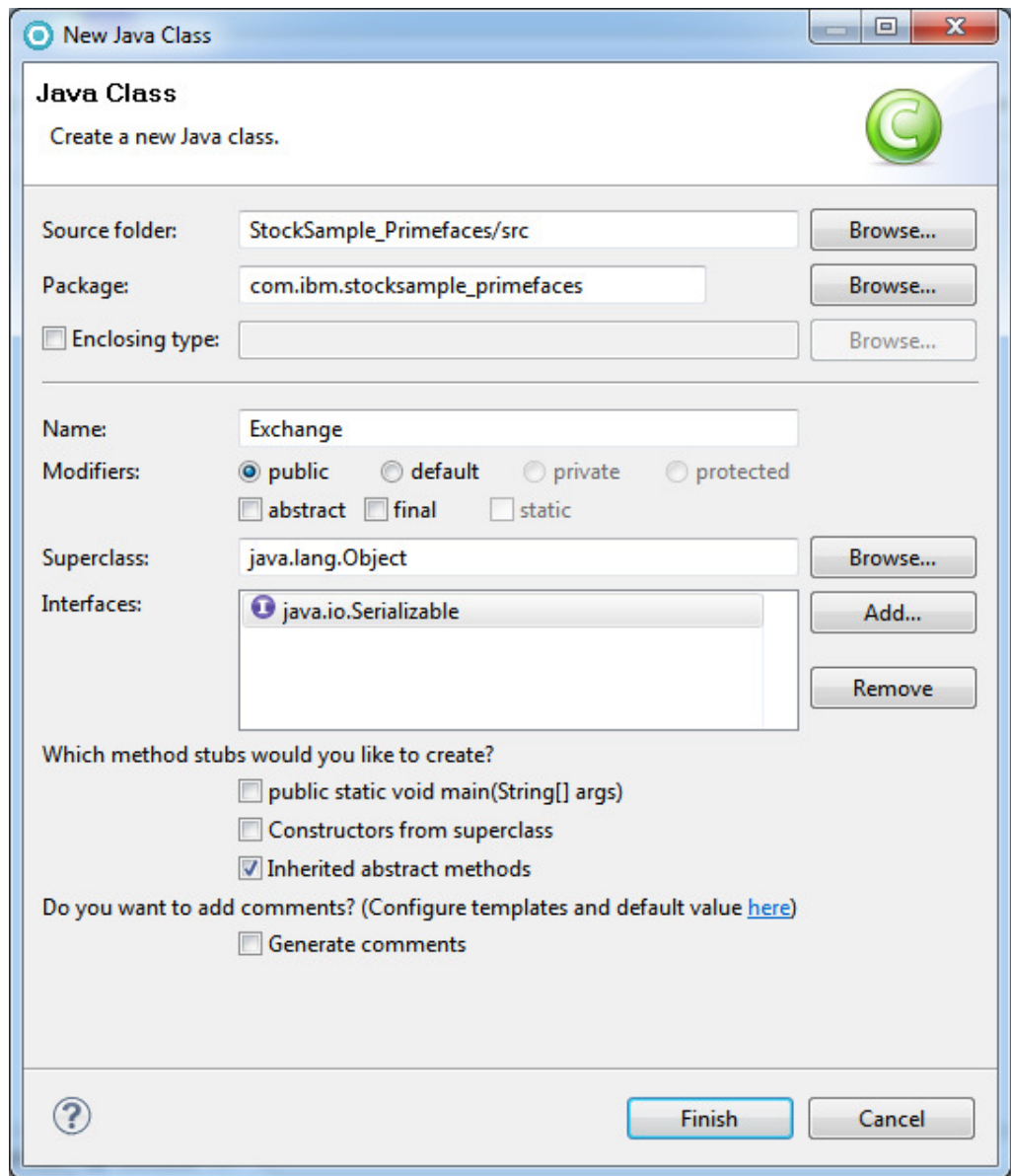
Creating scoped managed bean

Steps below illustrate the creation of managed bean using annotations in Rational Application Developer.

1. Create a Java class using **New->Class** from the context menu of **Java Resources** in the portlet project as shown in Figure 8. Give the name as **Exchange**.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Figure 8: Create a Managed Bean



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. Add the following annotations to the class, which define this class as a request scoped managed bean. The request scope managed bean must implement Serializable interface, as shown in Code Snippet 1:

Code Snippet 1

```
import java.io.Serializable;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

@ManagedBean(name="exchange")
@RequestScoped
public class Exchange implements Serializable {
```

3. Now add the class variables and constants for the managed bean, as shown in Code Snippet 2:

Code Snippet 2

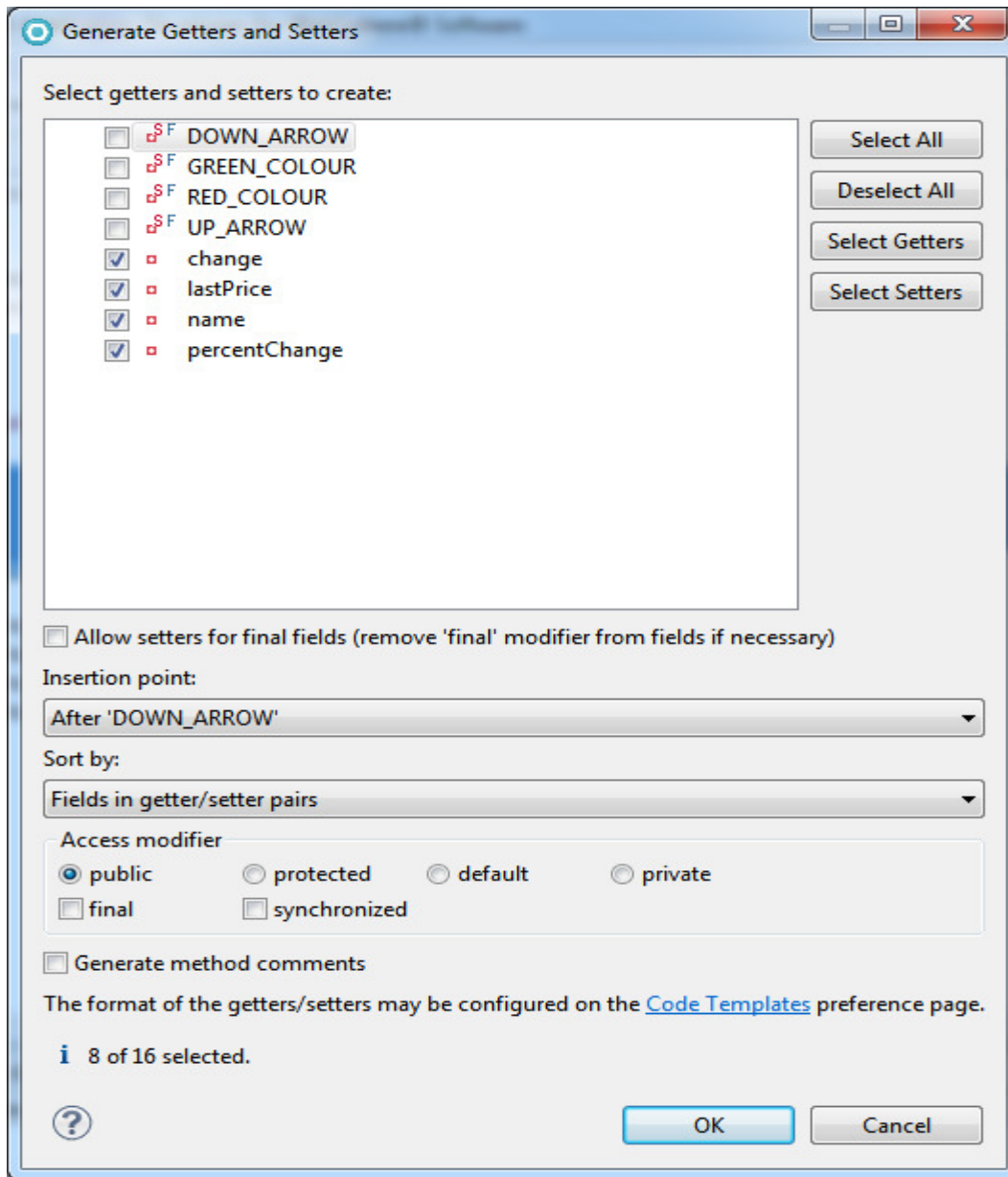
```
private String name;
private float lastPrice;
private String change;
private String percentChange;

private static final String RED_COLOUR = "#FF0000";
private static final String GREEN_COLOUR = "#006400";
private static final String UP_ARROW = "arrow_up_green.gif";
private static final String DOWN_ARROW = "arrow_down_red.gif";
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

- Right-click in the file and select **Source->Generate Getters and Setters**, to generate the getters and setters, as shown in Figure 9.

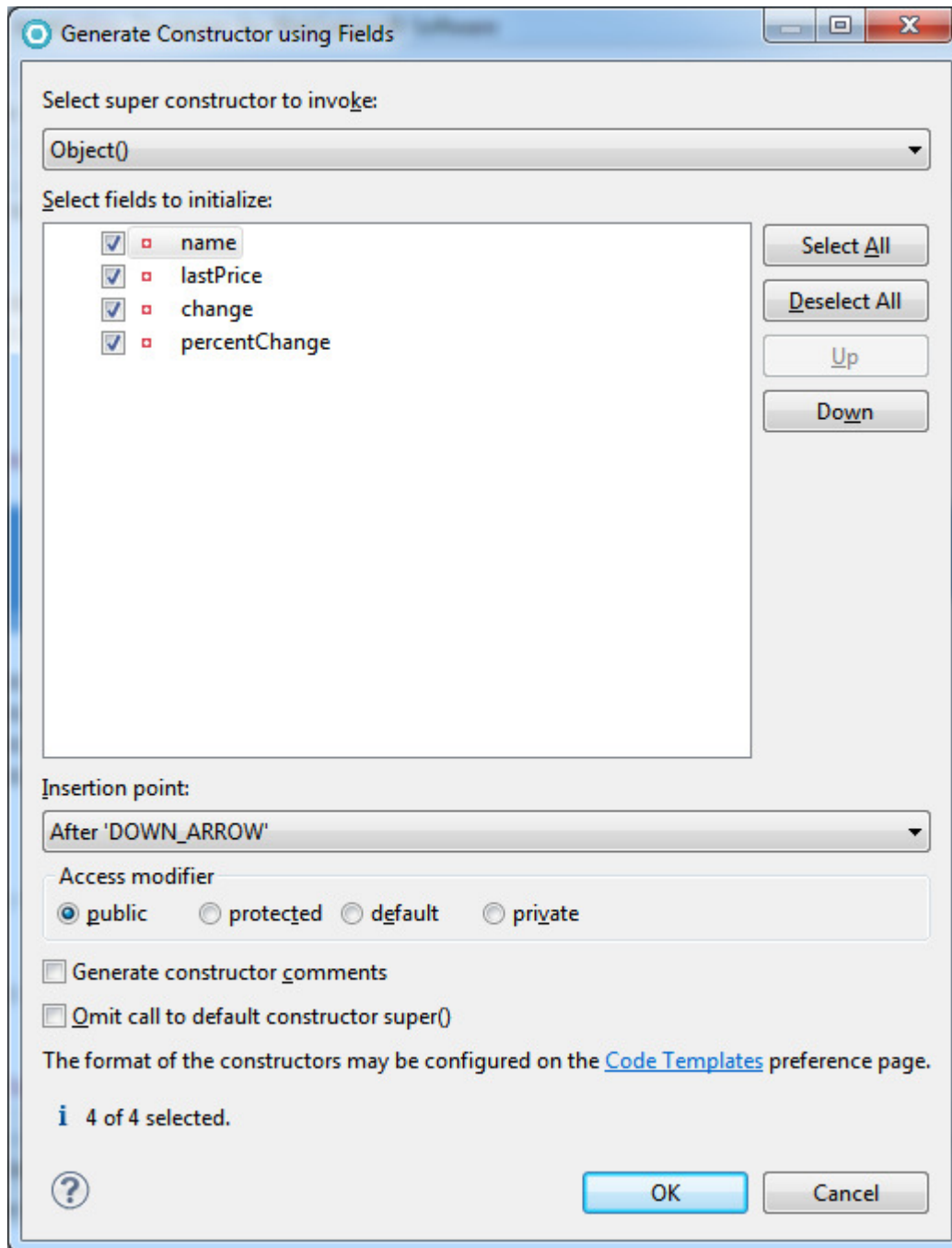
Figure 9: Generate Getters and Setters



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

5. Right-click in the file and select **Source->Generate Constructor using Fields** to generate the constructor with fields for managed bean, as shown in Figure 10.

Figure 10: Generate Constructor using Fields



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Custom code specific to the application

To customize the portlets and add exchange specific display to the Stock application, add the following methods:

1. Add `getPercentChangeFontColor()`, `getImage()` methods in the managed bean class, that is **Exchange.java**, which is needed to get the color of the arrow showing the trends, font color based on profit and loss in the change column. See **Exchange.java** (in the attached sample) for the details of this method.
2. Similarly create another annotated managed bean and specify the name as **ExchangeList**.
3. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *exchangeList*.
4. Add the following variables in the **ExchangeList.java file**, as shown in Code Snippet 3:

Code Snippet 3

```
private ArrayList<Exchange> exchange;  
private CartesianChartModel linearModel;  
private static final String EXCHANGE_FILE = "/exchange.xml";
```

5. Add `getExchange()`, `setStocks()`, `getLinearModel()`, `setLinearModel()`, `updateLinearModel()` methods in **ExchangeList.java** to display the default and updated values in the lineChart. See the **ExchangeList.java file** (in the attached sample) for the details of these methods.
6. Add `parseXML()` method to read the sample data from the XML file. See the **ExchangeList.java file** (in the attached sample) for the details of this method.

Developing GUI using Primefaces widgets

1. Open the **MarketWatchView.xhtml** file and drag **lineChart** item from the **primefaces** drawer of the **Palette** view onto the facelet (**MarketWatchView.xhtml**) file. A dialog box appears asking to copy new project resources.
2. Click **OK** to add Primefaces into the portlet project. Note that the `lineChart` tag is added to the page.
3. Right-click the linechart and select **Properties**. Specify the value of the **linechart** by clicking the browse button in the value field. Select “`{exchangeList.linearModel}`” as value of the linechart, as shown in Figure 11:

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Figure 11: lineChart Properties

Attribute Name	Value
binding	
breakOnNull	
enhancedLegend	
fill	
fillToZero	
id	lineChartRenderer1
legendPosition	e
maxX	
maxY	
minX	
minY	
rendered	
seriesColors	
shadow	
showMarkers	
stacked	
style	width: 435px;height:300px
styleClass	lineChart
title	
value	#{exchangeList.linearModel}
widgetVar	
xaxisAngle	
xaxisLabel	
yaxisAngle	
yaxisLabel	

The linechart tag is as shown in Code Snippet 4:

Code Snippet 4

```
<primefaces:lineChart styleClass="lineChart" id="lineChartRenderer1"
value="#{exchangeList.linearModel}" legendPosition="e"
style="width: 435px;height:300px"></primefaces:lineChart>
```

4. Drag the item **dataTable** from **Primefaces** drawer of the **Palette** view onto the facet (MarketWatchView.xhtml) file, and add columns as shown in Code Snippet 5:

Code Snippet 5

```
<primefaces:dataTable id="dataTableRenderer1"
var="exchange" value="#{exchangeList.exchange}" style="width: 435px">
  <primefaces:column styleClass="column" id="column1" headerText="Exchange">
    <primefaces:commandLink styleClass="commandLink"
id="commandLinkRenderer1" style="font-weight: bold"
update=":subview1:form1:chart1"
actionListener="#{exchangeList.updateLinearModel}"
value="#{exchange.name}">
      <f:param name="name" value="#{exchange.name}"></f:param>
    </primefaces:commandLink>
  </primefaces:column>
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

```
</primefaces:dataTable>
```

5. Drag **column** item from the **primefaces** drawer of the **Palette** view onto the facetlet (MarketWatchView.xhtml) file to add columns - Last Index, Change, and % Change, as shown in Code Snippet 6:

Code Snippet 6

```
<primefaces:column styleClass="column" id="column2" headerText="Last Index">
    <h:outputText styleClass="outputText" id="text1"
        value="#{exchange.lastPrice}" style="float: right;color:
            #{exchange.percentChangeFontColor}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column3" headerText="Change">
    <primefaces:graphicImage styleClass="graphicImage"
        id="graphicImageRenderer1" value="/resources/img/#{exchange.image}"
        style="float: right"></primefaces:graphicImage>
    <h:outputText styleClass="outputText" id="text2" value="#{exchange.change}"
        style="float: right;color:#{exchange.percentChangeFontColor}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column4" headerText="% Change">
    <h:outputText styleClass="outputText" id="text3"
        value="#{exchange.percentChange}%" style="float: right;color:
            #{exchange.percentChangeFontColor}"></h:outputText>
</primefaces:column>
```

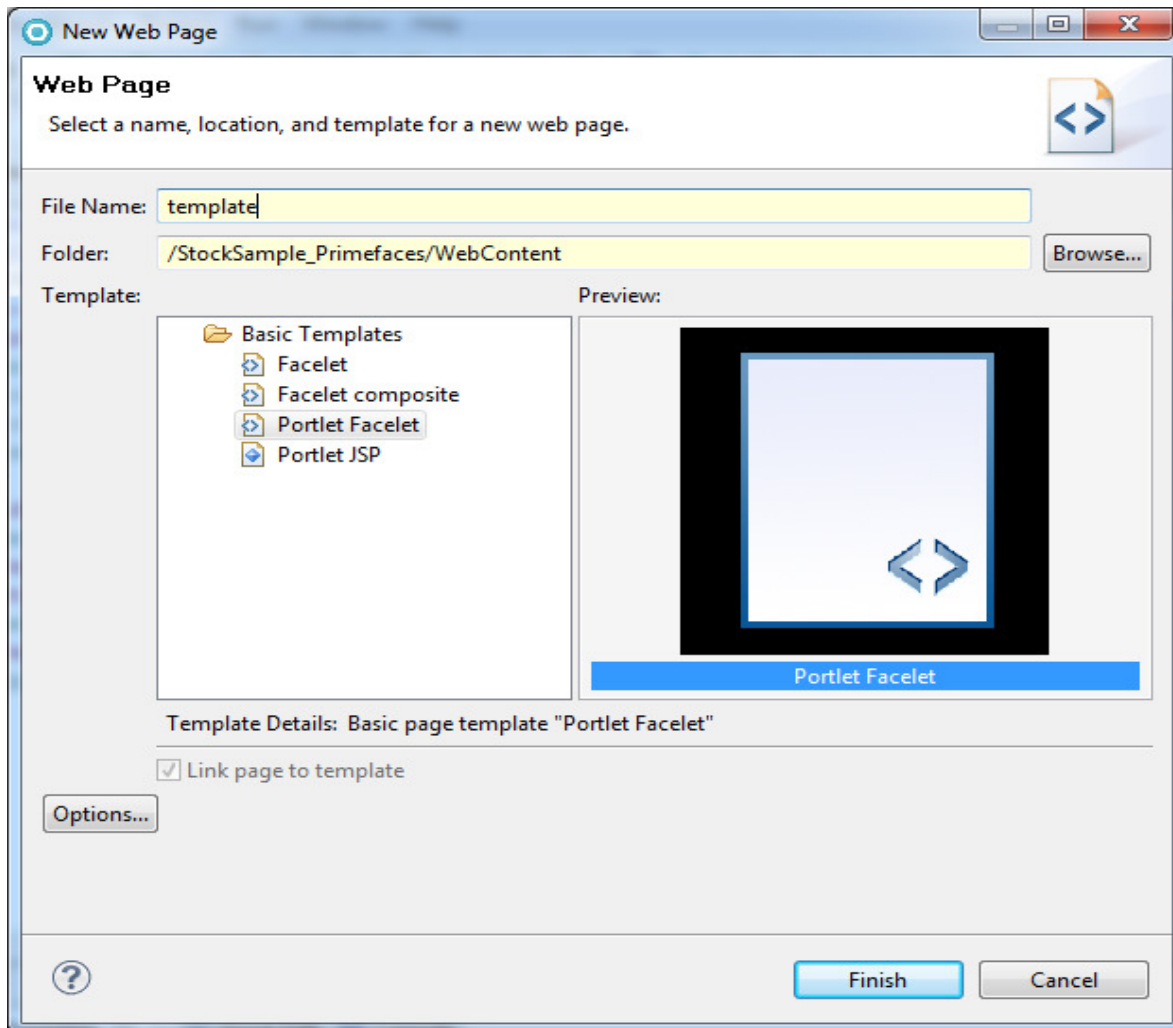
The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Developing Trading Portlet

This portlet is used to get the quotes for the stocks.

1. Select **New->Web Page** from the context menu of the portlet project.
2. Select **Portlet Facelet** in **New Web Page** wizard. Enter the **File Name** as **template** as shown in Figure 12.
3. Click **Finish**.

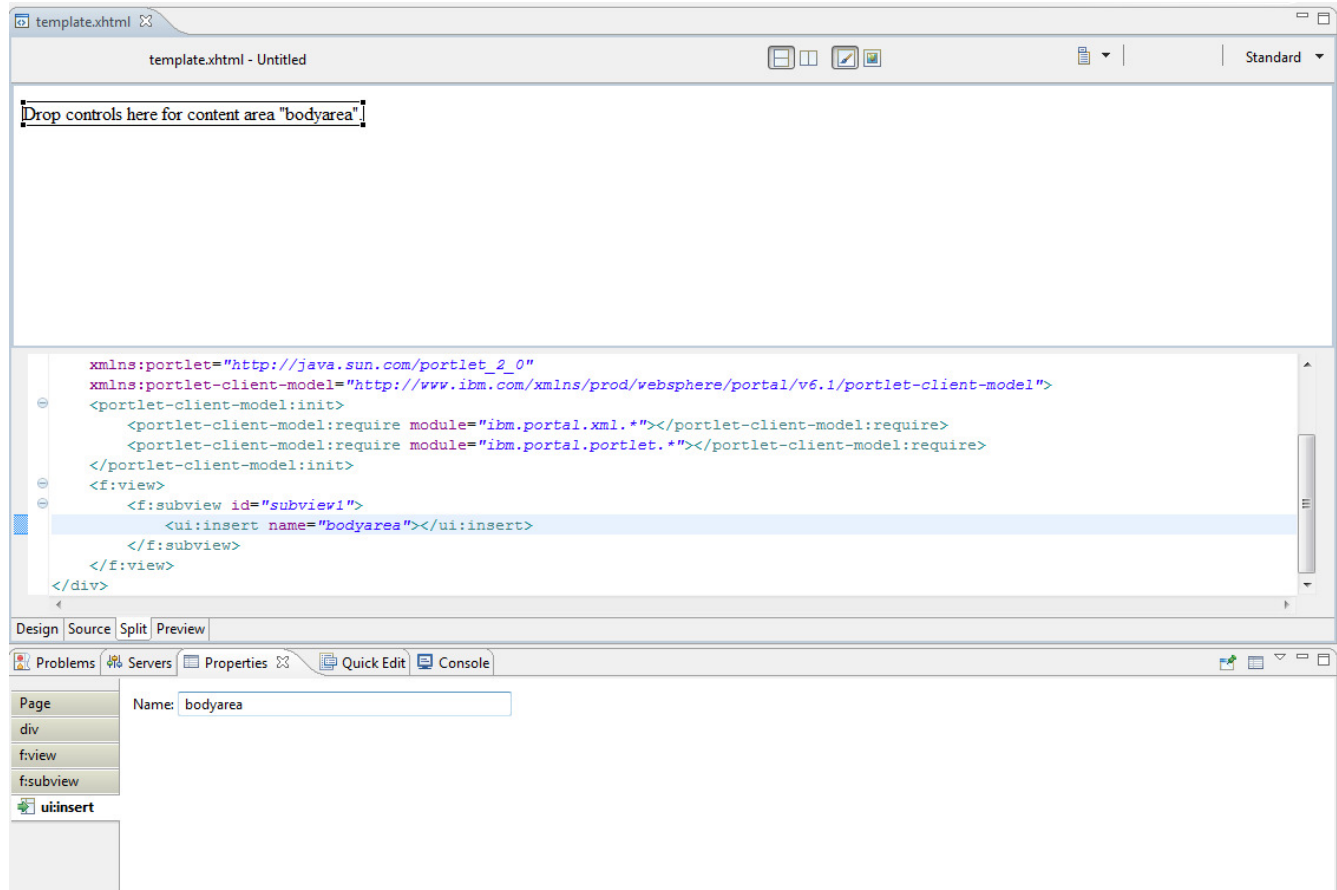
Figure 12: New Web Page Wizard to add template



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

4. Drag the **Insert** from the **Facelet** drawer of the Palette view onto the facelet (template.xhtml) file. Enter the name as **bodyarea** for the insert tag in the Properties view as shown in Figure 13:

Figure 13: Template Page

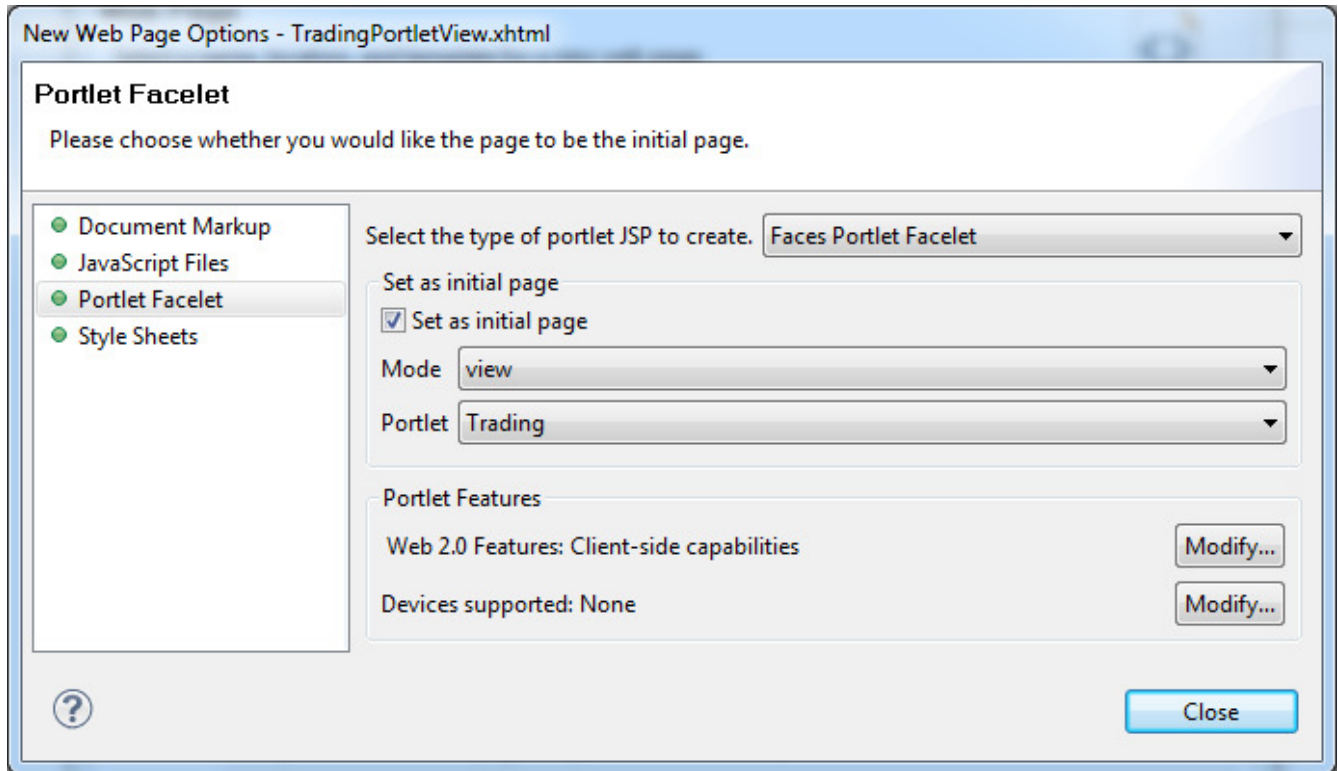


5. Select **New->Web Page** from the context menu of the portlet project and click **Options** and create a new portlet facelet page with the template you created.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

6. Select **Portlet Facelet** in **New Web Page Options** wizard and select **Set as initial page**. Click **Close**, as shown in Figure 14.

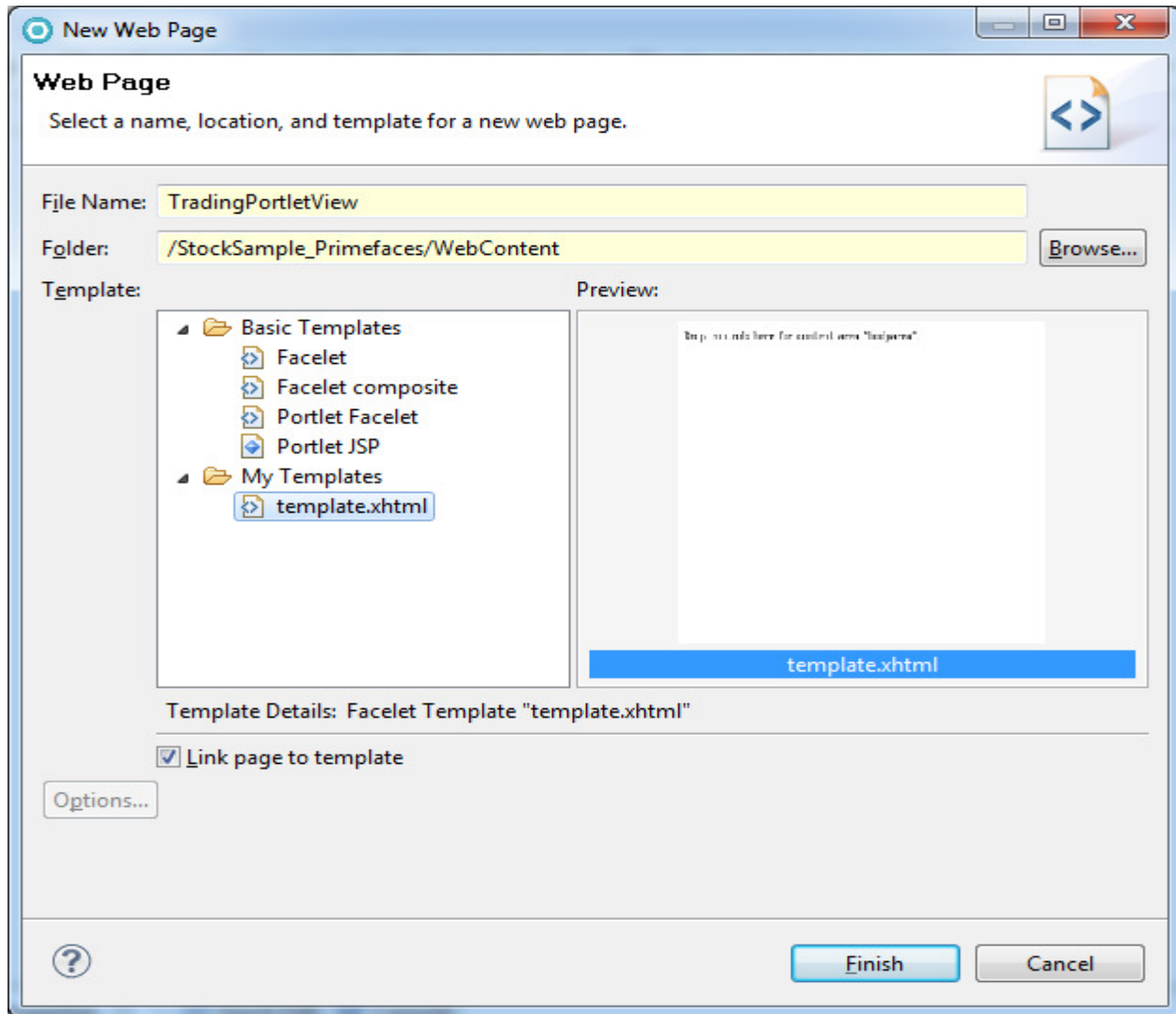
Figure 14: New Web Page Options



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

7. Select **template.xhtml** file from *MyTemplates*. Click **Finish**, as shown in Figure 15:

Figure 15: TradingPortletView Page



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Creating scoped managed bean

1. Create a Java class as shown in Figure 8. Specify the name as **TradingService**.
2. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *tradingService*.
3. Add the class variables and constants for the managed bean, as shown in Code Snippet 7:

Code Snippet 7

```
@ManagedProperty(value = "#{param.exchangename}")
private String exchangename;

@ManagedProperty(value = "#{param.stockname}")
private String stockname;

@ManagedProperty(value = "#{param.portletname}")
private String portletname;

private String lastPrice;
private String percentChange;
private String dataExists = "hidden";
private String message;

private static final String PORTFOLIO_FILE = "/portfolio.xml";
private static final String TRADING_FILE = "/trading.xml";
private static final String WATCHLIST_FILE = "/watchlist.xml";
```

4. Right-click in file and select **Source->Generate Getters and Setters**. to generate the getters and setters for the added classes, as shown in Figure 9.

Custom code specific to the application

To customize the portlets and add trading specific service to the Stock application, add the following methods. See the **TradingService.java** (in the attached sample) for the details about the added methods.

1. Add getQuotes(), getSelectedStock() methods to get the data for the selected stock. See
2. Add buyPage(), sellPage(), goBack() methods to navigate to Buy and Sell page respectively.
3. Add buyStock() method to buy the stock and add details of that stock in XML file.
4. Similarly add sellStock() method to sell the stock and update details of that stock in XML file.
5. Now add addToWatch() method to add the selected stocks in your watchlist.
6. Create a Java class as shown in Figure 8. Specify the name as **StockBean**.
7. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *stockBean*.
8. Now add the class variables for the managed bean, as shown in Code Snippet 8:

Code Snippet 8

```
private String exchangename;
private String stockname;
private String lastPrice;
private String percentChange;
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

9. Right-click in the file and select **Source->Generate Getters and Setters**, to generate the getters and setters, as shown in Figure 9.
10. Right-click in the file and select **Source->Generate Constructor using Fields** to generate the constructor with fields for managed bean, as shown in Figure 10. See the code generated in Code Snippet 9:

Code Snippet 9

```
public StockBean(String exchangename, String stockname, String lastPrice,
                String percentChange) {
    this.exchangename = exchangename;
    this.stockname = stockname;
    this.lastPrice = lastPrice;
    this.percentChange = percentChange;
}
```

11. Add `getImage()`, `getPercentChangeFontColor()` methods in the managed bean class that is, **StockBean.java**, needed to get the color of the arrow showing the trends, font color based on profit and loss in the *% change* and *last price* columns. See **StockBean.java** (in the attached sample) for the details of this method.
12. Create a Java class as shown in Figure 8. Specify the name as **BuySellBean**.
13. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *buySellBean*.
14. Now add the class variables for the managed bean, as shown in Code Snippet 10:

Code Snippet 10

```
@ManagedProperty(value = "#{param.exchangename}")
private String exchangename;

@ManagedProperty(value = "#{param.stockname}")
private String stockname;

@ManagedProperty(value = "#{param.price}")
private String price;

@ManagedProperty(value = "#{param.quantity}")
private String quantity;

@ManagedProperty(value = "#{param.portletname}")
private String portletname;

private String visibility = "visible";
```

15. Right-click in the file and select **Source->Generate Getters and Setters** to generate the getters and setters as shown in Figure 9.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Custom code specific to the application

To customize the portlets and add stock trade confirmation services to the Stock application, add `buyConfirmPage()`, `sellConfirmPage()` methods to navigate to the confirmation page of buy and sell operations. See `BuySellBean.java` (in the attached sample) for the details of this method.

Developing GUI using Primefaces widgets

In this application composite component helps to display faces messages in the page. You can add the composite component to different pages, wherever these messages are required. To add composite components:

1. Create a **Facelet composite** page, and name it as **message.xhtml**, as shown in Figure 12.
2. Drag **Output** from the **Standard Faces Components** drawer of the **Palette** view onto the facelet(`message.xhtml`), and set the properties as shown in Figure 16:

Figure 16: Properties view Facelet Composite

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:composite="http://java.sun.com/jsf/composite"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">

  <composite:interface></composite:interface>
  <composite:implementation>
    <h:outputText styleClass="outputText" id="text1"
      value="#{tradingService.message}"
      style="font-style: italic; color: red"></h:outputText>
  </composite:implementation>
</html>
```

The Properties view for the selected `h:outputText` widget shows the following configuration:

- Id:** text1
- Format:** String
- Value:** #{tradingService.message}
- Render the HTML in the value
- Style Props:** font-style: italic; color: red
- Classes:** outputText

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

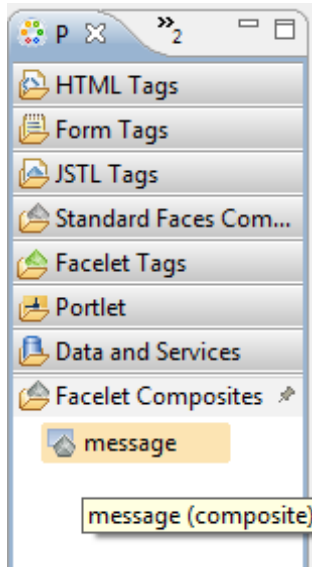
IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

3. Open the **TradingPortletView.xhtml** file, and insert the composite component. Drag **message** from the **Facelet Composites** drawer of the **Palette** view onto the facelet, as shown in Figure 17:

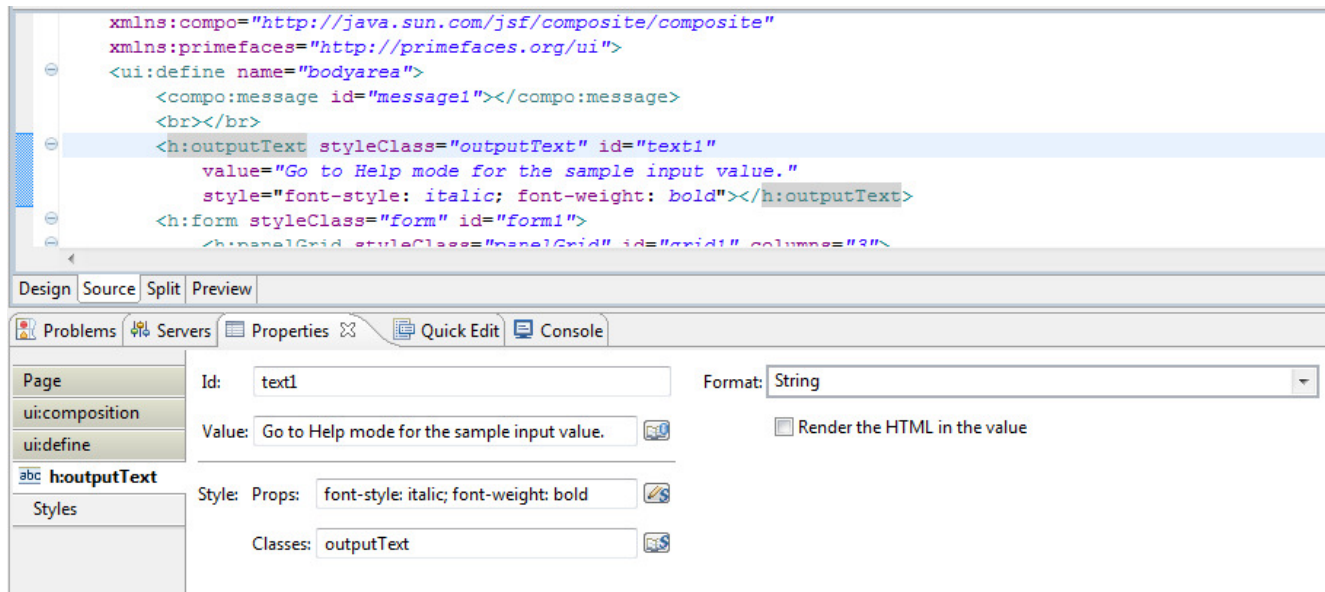
Figure 17: Facelet Composite in palette



4. Drag **Output** from **Standard Faces Components** drawer of the **Palette** view onto the facelet (**TradingPortletView.xhtml**) and set the properties, as shown in Figure 18:

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

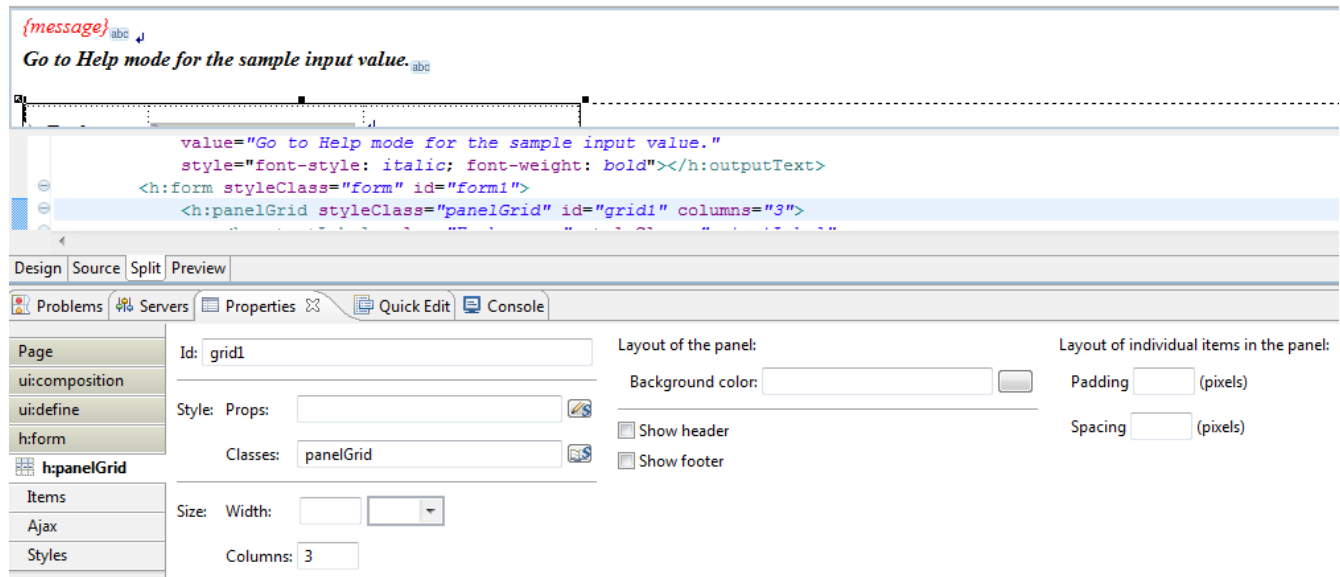
Figure 18: Properties view output text



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

5. Drag **Panel Grid** from **Standard Faces Components** drawer of the **Palette** view onto the facetlet (TradingPortletView.xhtml) and set the properties, as shown in Figure 19:

Figure 19: Panel Grid



6. Drag **Label** from **Standard Faces Components** drawer of the **Palette** view inside panel grid onto the facetlet (TradingPortletView.xhtml) and name it as **Exchange**.
7. Drag **selectOneMenu** from **primefaces** drawer of the **Palette** view inside panel grid onto the facetlet (TradingPortletView.xhtml). Set the choices and the **Value**. It is needed to view the list of exchanges, to select from, as shown in Code Snippet 11:

Code Snippet 11

```
<h:outputLabel value="Exchange: " styleClass="outputLabel"
id="labell" for="selectOneMenuRenderer1"
style="font-weight: bold; font-size: small"></h:outputLabel>
<primefaces:selectOneMenu styleClass="selectOneMenu"
id="selectOneMenuRenderer1" value="#{tradingService.exchangename}">
  <f:selectItem itemLabel="SAUAGA" itemValue="SAUAGA" id="selectItem1" />
  <f:selectItem itemLabel="JASSIN" itemValue="JASSIN" id="selectItem2" />
  <f:selectItem itemLabel="MANANE" itemValue="MANANE" id="selectItem3" />
  <f:selectItem itemLabel="MANGAB" itemValue="MANGAB" id="selectItem4" />
  <f:selectItem itemLabel="MEGKUC" itemValue="MEGKUC" id="selectItem5" />
</primefaces:selectOneMenu>
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.
 IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING,
 BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or
 exclusions may not apply to you. IBM shall not be liable
 for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

8. Drag **Label** named **Stock:** from **Standard Faces Components** drawer of the **Palette** view onto the facetlet (TradingPortletView.xhtml) and add a **inputText** from **primefaces** drawer under the **Palette** view with the properties set as shown in Figure 20:

Figure 20: TradingService stockname properties

The screenshot shows an IDE window with two panes. The top pane displays XML code for a PrimeFaces inputText component. The bottom pane shows the Properties view for the selected component.

```

</primefaces:selectOneMenu>
<br></br>
<h:outputLabel value="Stock: " styleClass="outputLabel" id="label2"
  for="inputTextRenderer1"
  style="font-weight: bold; font-size: small"></h:outputLabel>
<primefaces:inputText styleClass="inputText" id="inputTextRenderer1"
  value="#{tradingService.stockname}"></primefaces:inputText>
  
```

The Properties view shows the following table:

Attribute Name	Value
converter	
converterMessage	
dir	
disabled	
id	inputTextRenderer1
immediate	
label	
lang	
maxlength	
readonly	
rendered	
required	
requiredMessage	
size	
style	
styleClass	inputText
tabindex	
title	
type	
validator	
validatorMessage	
value	#{tradingService.stockname}
valueChangeListener	

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

9. Drag **commandButton**, next to **Stock** to get the quote based on the stock entered. Set the **actionListener** of the **commandButton** to the **getQuotes** method, as shown in Figure 21:

Figure 21: Command Button Properties view

```
<primefaces:commandButton styleClass="commandButton"
    id="commandButtonRenderer1" value="Get Quotes"
    update=":subview1:message1:text1,dataTableRenderer1"
    actionListener="#{tradingService.getQuotes}">
  <f:param name="exchangenam" value="#{tradingService.exchangenam}"></f:param>
  <f:param name="stockname" value="#{tradingService.stockname}"></f:param>
</primefaces:commandButton>
```

Attribute Name	Value
actionListener	#{tradingService.getQuotes}
ajax	
alt	
async	
binding	
dir	
disabled	
global	
icon	
iconPos	
id	commandButtonRenderer1
image	
immediate	
inline	
label	
lang	
process	
readonly	
rendered	
style	
styleClass	commandButton
tabindex	
title	
type	

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

10. Drag **dataTable** item from the **primefaces** drawer of the **Palette** view onto the facetlet (TradingPortletView.xhtml), and add the columns as shown in Code Snippet 12:

Code Snippet 12

```
<primefaces:dataTable id="dataTableRenderer1"
  var="stock" value="#{tradingService.selectedStock}"
  style="margin-left: 5px; width: 450px; visibility:
  #{tradingService.dataExists}">
  <primefaces:column styleClass="column" id="column1" headerText="Name">
    <h:outputText styleClass="outputText" id="text2"
    value="#{stock.stockname}"></h:outputText>
  </primefaces:column>
</primefaces:dataTable>
```

11. Drag **column** item from the **primefaces** drawer of the **Palette** view onto the facetlet (TradingPortletView.xhtml) file to add columns - Last Price and % Change, as shown in Code Snippet 13:

Code Snippet 13

```
<primefaces:column styleClass="column" id="column2" headerText="% Change">
  <h:outputText styleClass="outputText" id="text3"
  value="#{stock.percentChange}%"
  style="float: right;color:#{stock.percentChangeFontColor}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column3" headerText="Last Price">
  <primefaces:graphicImage id="graphicImageRenderer1"
  value="/resources/img/#{stock.image}" style="float:
  right"></primefaces:graphicImage>
  <h:outputText styleClass="outputText" id="text4"
  value="#{stock.lastPrice}" style="float:
  right;color:#{stock.percentChangeFontColor}"></h:outputText>
</primefaces:column>
```

12. Add three columns to the data table with command links to buy, sell, and watch the stocks, as shown in Code Snippet 14:

Code Snippet 14

```
<primefaces:column styleClass="column" id="column4" headerText=" ">
  <primefaces:commandLink styleClass="commandLink"
  id="commandLinkRenderer1" ajax="false"
  style="float: right;font-weight: bold" action="#{tradingService.buyPage}"
  value="Buy">
    <f:param name="exchangenname" value="#{stock.exchangenname}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
    <f:param name="portletname" value="trading"></f:param>
  </primefaces:commandLink>
</primefaces:column>
<primefaces:column styleClass="column" id="column5" headerText=" ">
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

```

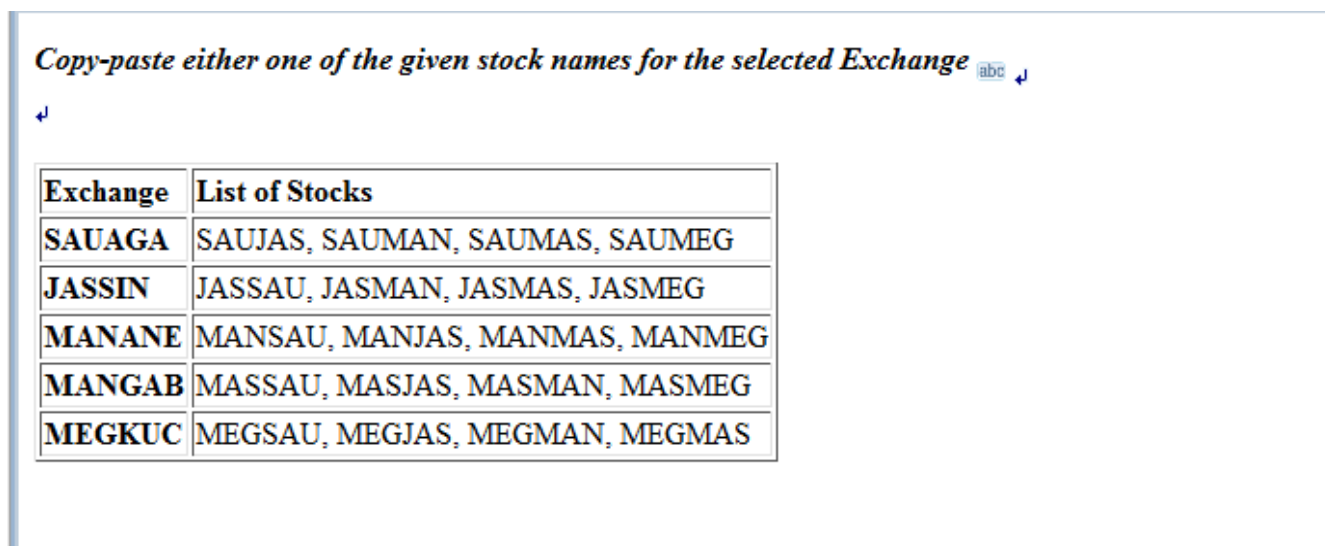
<primefaces:commandLink styleClass="commandLink"
id="commandLinkRenderer2" ajax="false"
style="float: left;font-weight: bold" action="#{tradingService.sellPage}"
value="Sell">
    <f:param name="exchangename" value="#{stock.exchangename}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
    <f:param name="portletname" value="trading"></f:param>
</primefaces:commandLink>
</primefaces:column>
<primefaces:column styleClass="column" id="column6" headerText=" ">
    <primefaces:commandLink styleClass="commandLink"
id="commandLinkRenderer3" ajax="false"
style="float: left;font-weight: bold" action="#{tradingService.addToWatch}"
value="Watch">
    <f:param name="exchangename" value="#{stock.exchangename}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
    </primefaces:commandLink>
</primefaces:column>

```

13. Add **Help mode** to display the list of stock and exchange names that are used to get the quotes in the **TradingPortletView** page. For this, create a **Portlet Facelet** page name as **TradingHelp.xhtml**, as shown in Figure 12.

14. Add content same as shown in the Figure 22. See the sample for details of this page.

Figure 22: Trading Help page content



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

15. Add a parameter **init-param** to the **Initialization** section under the **Portlets** tab for **Trading** portlet as shown in Figure 23:

Figure 23: Portlet tab Initialization section

The screenshot shows the 'Portlets' tab in a configuration tool. On the left, a list of portlets includes 'Market Watch', 'Trading' (selected), 'Portfolio', and 'Watchlist'. Below this list are buttons for 'New...', 'Add...', 'Remove', and 'Edit...'. The main area is divided into two sections: 'Initialization' and 'Container Runtime Options'. The 'Initialization' section contains a list of initialization parameters: 'com.ibm.faces.portlet.page.view /TradingPortletView.xhtml', 'wps.markup.html', and 'com.ibm.faces.portlet.page.help /TradingHelp.xhtml'. To the right of this list are buttons for 'Add...', 'Edit...', and 'Remove'. The 'Container Runtime Options' section is currently empty and has 'Add...' and 'Edit...' buttons. At the bottom, a navigation bar includes tabs for 'Overview', 'Portlets', 'Events', 'Render Params', 'Filters', 'URL Listeners', 'Security', 'Modes', 'States', 'Attributes', 'Runtime Options', 'Miscellaneous', and 'Source'.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Developing Portfolio Portlet

The Portfolio portlet displays the stocks in a trading account portfolio, that is the list of the stocks that are bought. Create a managed bean for the Portfolio portlet to hold the required data.

Creating scoped managed bean

1. Create a Java class as shown in Figure 8. Give the name as **Portfolio**.
2. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *portfolio*.
3. Now add the class variables and constants for the managed bean, as shown in Code Snippet 15:

Code Snippet 15

```
private String exchangename;  
private String stockname;  
private String buyPrice;  
private String currentPrice;  
private String quantity;  
  
private static final String RED_COLOUR = "#FF0000";  
private static final String GREEN_COLOUR = "#006400";  
private static final String UP_ARROW = "arrow_up_green.gif";  
private static final String DOWN_ARROW = "arrow_down_red.gif";
```

4. Right-click in the file and select **Source->Generate Getters and Setters**, to generate the getters and setters, as shown in Figure 9.
5. Right-click in the file and select **Source->Generate Constructor using Fields** to generate the constructor with fields for managed bean, as shown in Figure 10.

Custom code specific to the application

To customize the portlets and add trading specific status services to the Stock application, add the following methods.

1. Add `getStatusImage()`, `getFontColor()`, `getMarketValue()`, `getStatus()` methods in the managed bean class that is, **Portfolio.java**, which is needed to get the color of the arrow showing the trends, font color based on profit and loss in the "Profit/Loss" column. See **Portfolio.java** (in the attached sample) for the details of these methods.
2. Similarly create another annotated managed bean and specify the name as **PortfolioList**.
3. Add annotations to the class, as shown in Code Snippet 1. Give managed bean name as *portfolioList*.
4. Add the following variables in **PortfolioList.java**, as shown in Code Snippet 16:

Code Snippet 16

```
private ArrayList<Portfolio> stocks;  
private ArrayList<Portfolio> filteredStocks;  
private String selectedView;  
private static final String PORTFOLIO_FILE = "/portfolio.xml";
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Custom code specific to the application

To customize the portlets and add portfolio specifics service to the Stock application, add the following methods. See the **PortfolioList.java** (in the attached sample) for the details about the added methods.

1. Add `updateValue()`, `getFilteredStocks()` methods which are needed to filter the stock data.
2. Add `parseXML()`, `checkPortfolioData()` methods which are needed to read the data from the XML file.

Developing GUI using Primefaces widgets

1. Open the **PortfolioView.xml** file and drag a **Panel-Grid** item from the **Standard Faces Components drawer** onto the facelet (`PortfolioView.xhtml`), as shown in Code snippet 17:

Code Snippet 17

```
<h:panelGrid styleClass="panelGrid" id="grid1" columns="2"></h:panelGrid>
```

2. Drag **Label** from the **Standard Faces Components drawer** of the **Palette** view inside panel grid onto the facelet (`PortfolioView.xhtml`). Specify the value as **Select View**: Following tag would be added in facelet, as shown in Code snippet 18:

Code Snippet 18

```
<h:outputLabel styleClass="outputLabel" value="Select View: "
id="labell" for="selectOneMenuRenderer1" style="font-weight: bold; font-size:
small"></h:outputLabel>
```

3. Drag **selectOneMenu** from **primefaces** drawer of the **Palette** view inside panel grid onto the facelet (`PortfolioView.xhtml`). Set the choices and the Value. This is needed to view the filtered data of stocks based on profit and loss, as shown in Code Snippet 19:

Code Snippet 19

```
<primefaces:selectOneMenu styleClass="selectOneMenu" id="selectOneMenuRenderer1"
value="#{portfolioList.selectedView}"
valueChangeListener="#{portfolioList.updateValue}">
  <f:selectItem itemLabel="All" itemValue="All" id="selectItem1" />
  <f:selectItem itemLabel="Profit" itemValue="Profit" id="selectItem2" />
  <f:selectItem itemLabel="Loss" itemValue="Loss" id="selectItem3" />
  <primefaces:ajax event="change" update="dataTableRenderer1" process="form1" />
</primefaces:selectOneMenu>
```

4. Drag **dataTable** item from the **primefaces** drawer of the **Palette** view onto the facelet (`PortfolioView.xhtml`) and add the columns, as shown in Code Snippet 20:

Code Snippet 20

```
<primefaces:dataTable id="dataTableRenderer1" var="stock"
value="#{portfolioList.filteredStocks}" style="width: 700px;margin-left:3px">
  <primefaces:column styleClass="column" id="column1" headerText="Exchange">
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

```

        <h:outputText styleClass="outputText" id="text1"
            value="#{stock.exchangename}"></h:outputText>
    </primefaces:column>
</primefaces:dataTable>

```

5. Drag **column** item from the **primefaces** drawer of the **Palette** view onto the facelet (PortfolioView.xhtml) file to add columns - Name, Buy Price, Current Price, Quantity, Market Value and Pofit/Loss, as shown in Code Snippet 21:

Code Snippet 21

```

<primefaces:column styleClass="column" id="column2" headerText="Name">
    <h:outputText styleClass="outputText" id="text2"
        value="#{stock.stockname}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column3" headerText="Buy Price">
    <h:outputText styleClass="outputText" id="text3"
        style="float: right" value="#{stock.buyPrice}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column4" headerText="Current Price">
    <h:outputText styleClass="outputText" id="text4"
        style="float: right" value="#{stock.currentPrice}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column5" headerText="Quantity">
    <h:outputText styleClass="outputText" id="text5"
        style="float: right" value="#{stock.quantity}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column6" headerText="Market Value">
    <h:outputText styleClass="outputText" id="text6"
        style="float: right" value="#{stock.marketValue}"></h:outputText>
</primefaces:column>
<primefaces:column styleClass="column" id="column7" headerText="Profit/Loss">
    <primefaces:graphicImage styleClass="graphicImage"
        id="graphicImageRenderer1" value="/resources/img/#{stock.statusImage}"
        style="float: right"></primefaces:graphicImage>
    <h:outputText styleClass="outputText" id="text7" style="float: right;color:
        #{stock.fontColor}" value="#{stock.status}"></h:outputText>
</primefaces:column>

```

6. Add 2 more columns to the data table with command links to buy and sell the stocks, as shown in Code Snippet 22:

Code Snippet 22

```

<primefaces:column styleClass="column" id="column8" headerText=" ">
    <primefaces:commandLink styleClass="commandLink"
        id="commandLinkRenderer2" ajax="false"
        style="float: right;font-weight: bold"
        action="#{tradingService.buyPage}" value="Buy">
        <f:param name="exchangename" value="#{stock.exchangename}"></f:param>
        <f:param name="stockname" value="#{stock.stockname}"></f:param>
    </primefaces:commandLink>

```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

```

        <f:param name="portletname" value="portfolio"></f:param>
    </primefaces:commandLink>
</primefaces:column>
<primefaces:column styleClass="column" id="column9" headerText=" ">
    <primefaces:commandLink styleClass="commandLink"
        id="commandLinkRenderer3" ajax="false"
        style="float: left;font-weight: bold"
        action="#{tradingService.sellPage}" value="Sell">
        <f:param name="exchangenname" value="#{stock.exchangenname}"></f:param>
        <f:param name="stockname" value="#{stock.stockname}"></f:param>
        <f:param name="portletname" value="portfolio"></f:param>
    </primefaces:commandLink>
</primefaces:column>

```

Developing Watchlist portlet

The Watchlist portlet displays a list of stocks that are on the watch list of a trading account. You can buy or sell these stocks, or remove them from the watch list.

Creating scoped managed bean

1. Create a Java class as shown in Figure 8. and name it as **Watch**.
2. Add annotations to the class, as shown in Code Snippet 1, and name the managed bean as *watch*.
3. Add the class variables and constants for the managed bean, as shown in Code Snippet 23:

Code Snippet 23

```

private String exchangenname;
private String stockname;
private float lastPrice;
private String percentChange;

private static final String RED_COLOUR = "#FF0000";
private static final String GREEN_COLOUR = "#006400";
private static final String UP_ARROW = "arrow_up_green.gif";
private static final String DOWN_ARROW = "arrow_down_red.gif";

```

4. Right-click in the file and select **Source->Generate Getters and Setters** to generate the getters and setters for added class, as shown in Figure 9.
5. Right-click in the file and select **Source->Generate Constructor using Fields** to generate the constructor with fields for managed bean, as shown in Figure 10.

Custom code specific to the application

To customize the portlets and add watch list specific display to the Stock application, add the following methods:

1. Add getImage(), getPercentChangeFontColor() methods in the managed bean class that is **Watch.java**, which is needed to get the color of the arrow showing the trends, font color based on profit and loss in the "last price" column. See **Watch.java** (in the attached sample) for the details of this method.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.

IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. Similarly create another annotated managed bean and specify the name as **WatchList**.
3. Add annotations to the class, as shown in Code Snippet 1, and name the managed bean as *watchList*.
4. Add the following variables in **WatchList.java**, as shown in Code Snippet 24:

Code Snippet 24

```
private ArrayList<Watch> stocks;
private static final String WATCHLIST_FILE = "/watchlist.xml";
private static Boolean ascendingExchange;
private static Boolean ascendingStock;
private static Boolean ascendingPercentChange;
private boolean deleteAction = false;
```

Custom code specific to the application

To customize the portlets and add watch list specific display to the Stock application, add the following methods. See the **WatchList.java** (in the attached sample) for the details about the added methods.

1. Add `sortByExchangeName()`, `sortByStockName()`, `sortByPercentChange()` methods in **WatchList.java** to sort the columns of the dataTable.
2. Add `parseXML()` method to read the sample data from the XML file.
3. Add `deleteStock()` method to remove the selected stock from the **watchlist** portlet.

Developing GUI using Primefaces widgets

1. Open **WatchListView.xhtml** and drag and drop **dataTable** under **primefaces** section of the **Palette** view in the facelet (**WatchListView.xhtml**) and add the columns, as shown in Code Snippet 25:

Code Snippet 25

```
<primefaces:dataTable id="dataTableRenderer1"
var="stock" value="#{watchList.stocks}" style="width: 500px">
  <primefaces:column id="column1" styleClass="column">
    <f:facet name="header">
      <primefaces:commandLink styleClass="commandLink" value="Exchange"
style="font-size: small; float: left" update="dataTableRenderer1"
actionListener="#{watchList.sortByExchangeName}">
    </primefaces:commandLink>
    </f:facet>
    <h:outputText id="text1" value="#{stock.exchangename}"></h:outputText>
  </primefaces:column>
</primefaces:dataTable>
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. Drag **column** item from the **primefaces** drawer of the **Palette** view onto the facelet (WatchListView.xhtml) file to add columns - Name, Last Price and % Change, as shown in Code Snippet 26:

Code Snippet 26

```
<primefaces:column id="column2" styleClass="column">
  <f:facet name="header">
    <primefaces:commandLink styleClass="commandLink" value="Name"
      style="font-size: small; float: left" update="dataTableRenderer1"
      actionListener="#{watchList.sortByStockName}">
    </primefaces:commandLink>
  </f:facet>
  <h:outputText id="text2" value="#{stock.stockname}"></h:outputText>
</primefaces:column>
<primefaces:column id="column3" styleClass="column">
  <f:facet name="header">
    <h:outputText value="Last Price" styleClass="outputText"
      style="font-size: small; float: right"></h:outputText>
  </f:facet>
  <primefaces:graphicImage id="graphicImageRenderer1"
    value="/resources/img/#{stock.image}" style="float:
    right"></primefaces:graphicImage>
  <h:outputText id="text3" value="#{stock.lastPrice}"
    style="float: right;color:#{stock.percentChangeFontColor}"></h:outputText>
</primefaces:column>
<primefaces:column id="column4" styleClass="column">
  <f:facet name="header">
    <primefaces:commandLink styleClass="commandLink" value="% Change"
      style="font-size: small; float: right" update="dataTableRenderer1"
      actionListener="#{watchList.sortByPercentChange}">
    </primefaces:commandLink>
  </f:facet>
  <h:outputText id="text4" style="float:
    right;color:#{stock.percentChangeFontColor}" value="#{stock.percentChange}
    %"></h:outputText>
</primefaces:column>
```

3. Add 3 more columns to the data table with command links which enable the user to buy, sell and unwatch the stocks, as shown in Code Snippet 27:

Code Snippet 27

```
<primefaces:column id="column5" headerText=" " styleClass="column">
  <primefaces:commandLink id="commandLinkRenderer2" ajax="false"
    style="float: right;font-weight: bold" styleClass="commandLink"
    action="#{tradingService.buyPage}" value="Buy">
    <f:param name="exchangenname" value="#{stock.exchangenname}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
    <f:param name="portletname" value="watchlist"></f:param>
  </primefaces:commandLink>
```

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

```

</primefaces:column>
<primefaces:column id="column6" headerText=" " styleClass="column">
  <primefaces:commandLink id="commandLinkRenderer3" ajax="false"
    style="float: left;font-weight: bold" styleClass="commandLink"
    action="#{tradingService.sellPage}" value="Sell">
    <f:param name="exchangenname" value="#{stock.exchangenname}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
    <f:param name="portletname" value="watchlist"></f:param>
  </primefaces:commandLink>
</primefaces:column>
<primefaces:column id="column7" headerText=" " styleClass="column">
  <primefaces:commandLink id="commandLinkRenderer4"
    styleClass="commandLink" update="dataTableRenderer1"
    style="float: left;font-weight: bold"
    actionListener="#{watchList.deleteStock}" value="Unwatch">
    <f:param name="exchangenname" value="#{stock.exchangenname}"></f:param>
    <f:param name="stockname" value="#{stock.stockname}"></f:param>
  </primefaces:commandLink>
</primefaces:column>

```

Similarly, create the following portlet facelets for buying and selling stocks, as shown in Figure 12. See sample code for details of these pages.

- Buy.xhtml
- BuyConfirm.xhtml
- BuySuccess.xhtml
- Sell.xhtml
- SellConfirm.xhtml
- SellError.xhtml
- SellSuccess.xhtml

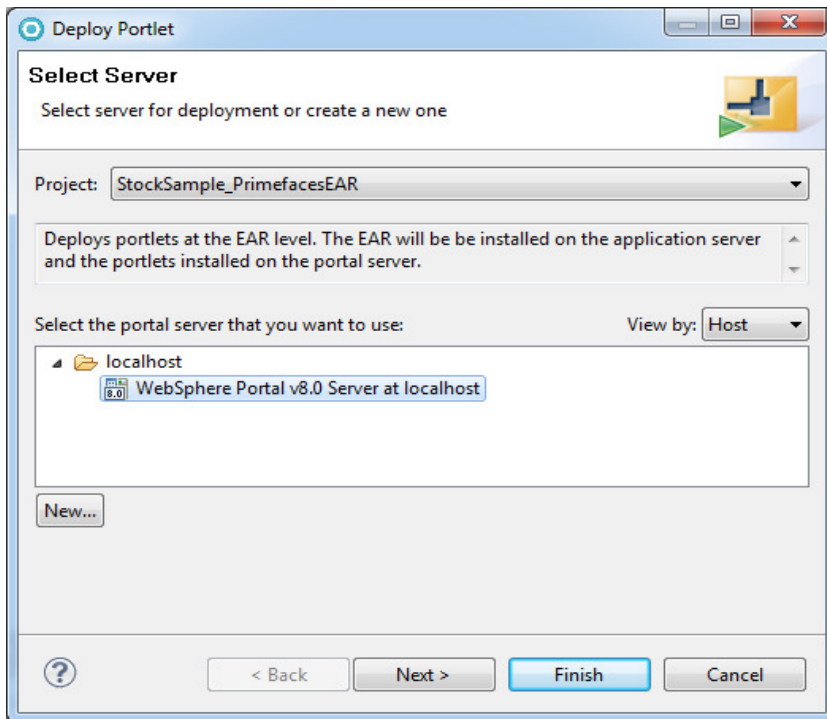
The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Deploying application on WebSphere Portal server

To deploy the Stock sample application:

1. Right-click on portlet project and select **Deploy Portlet..** **Deploy Portlet** wizard displays, as shown in Figure 24:

Figure 24: Deploy Portlet Wizard



2. Select **WebSphere Portal Server v8.0** and click **Finish**.
3. Right-click on server instance in the **Server** view and select **Open Portal**.
4. Click **Administration**. Select **Manage Pages** and click on **Content Root**.
5. Click the New Label and enter the "title" as **Bank Home** and Friendly URL name as **bankhome** and click OK.
6. Click the **Bank Home** label and create a new label title as **Investments** and Friendly URL name as **investments** and click OK.
7. Select **Investments** and click New Page. Create four pages as:
 - First Page – Name the title of the page as **Markets Overview** and Friendly URL name as **marketwatch**.
 - Second Page – Name the title of the page as **My Trading** and Friendly URL name as **trading**.
 - Third Page – Name the title of the page as **Portfolio** and Friendly URL name as **portfolio**.
 - Fourth Page – Name the title of the page as **Watchlist** and Friendly URL name as **watchlist**.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind.










IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

8. Select Edit Page Layout as shown in Figure 25 to add the following portlets deployed in step 3 to the corresponding pages:
 - For the first page, select the Market Watch portlet.

Figure 25: Edit Page Layout

Title	Unique name or Identifier	Status	
 Markets Overview	id:Z6_8P021800I00J40I4JAHS41GV2	Active	       

- For the second page, select the Trading portlet.
 - For the third page, select the Portfolio portlet.
 - For the fourth page, select the Watchlist portlet.
9. Click **Bank Home** from the Menu bar and click **Edit Mode**.
 10. Select **Layout** tab, select 1 column, click **Save**, and then go back to **View Mode**. Repeat the steps for My Trading, Portfolio and Watchlist pages also.

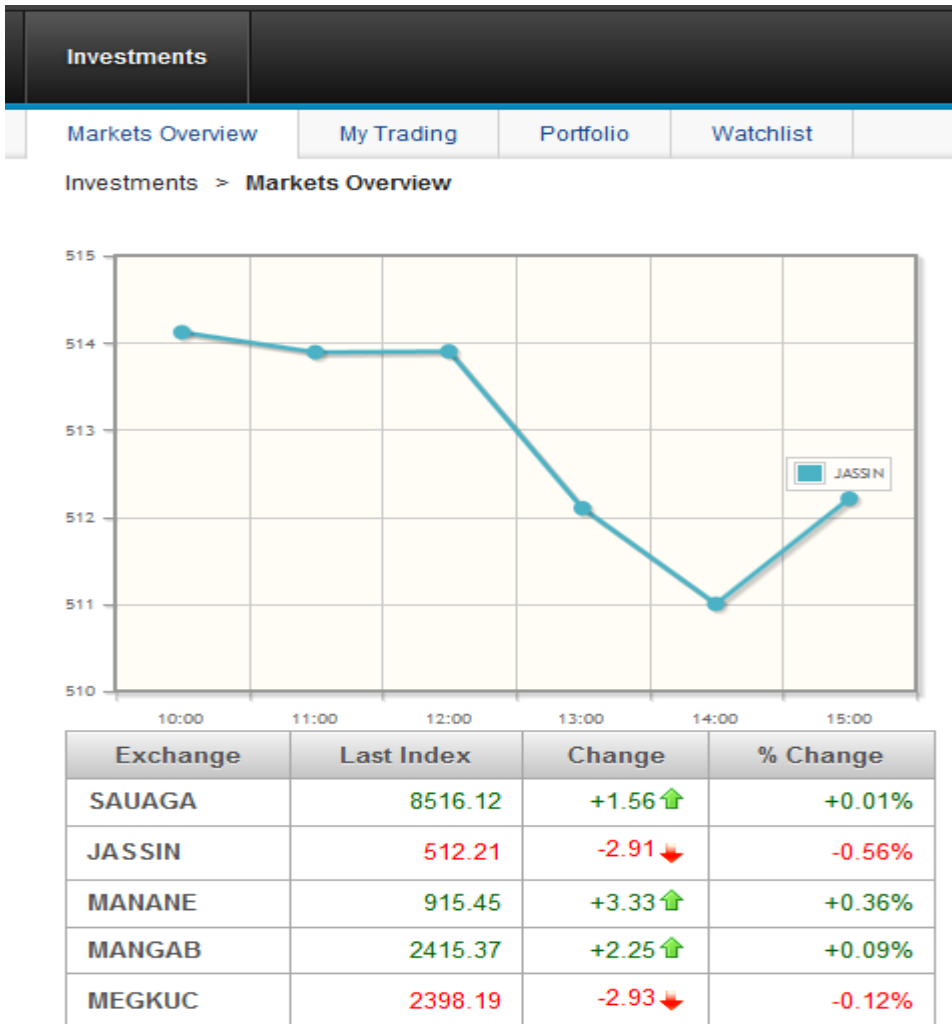
The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Executing the application on the server

Market Watch Portlet

Markets Overview page contains the Market Watch portlet to view the latest market trends. To see the graphs for different exchanges, click on any Exchange link. For example.: click JASSIN as shown in Figure 26:

Figure 26: Market Watch Portlet



NOTE: In this sample application the GREEN (upward arrow) color is used to show the upward trend of a stock and the RED (downward arrow) color is used to show the downward trend on any trading day. The green color depicts that the stock gained points from its last trading day, and the red color depicts that it lost points.

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Trading Portlet

Select any Exchange from the select box, enter stock name and click Get Quotes. A data table will display the information about that stock as shown in Figure 27:

Figure 27: Trading Portlet

The screenshot shows the 'Investments' section with a navigation bar containing 'Markets Overview', 'My Trading', 'Portfolio', and 'Watchlist'. Below the navigation bar, the breadcrumb 'Investments > My Trading' is visible. The main content area contains the text 'Go to Help mode for the sample input value.' followed by an 'Exchange:' dropdown menu set to 'SAUAGA'. Below that is a 'Stock:' text input field and a 'Get Quotes' button.

Figure 28: Trading Portlet View

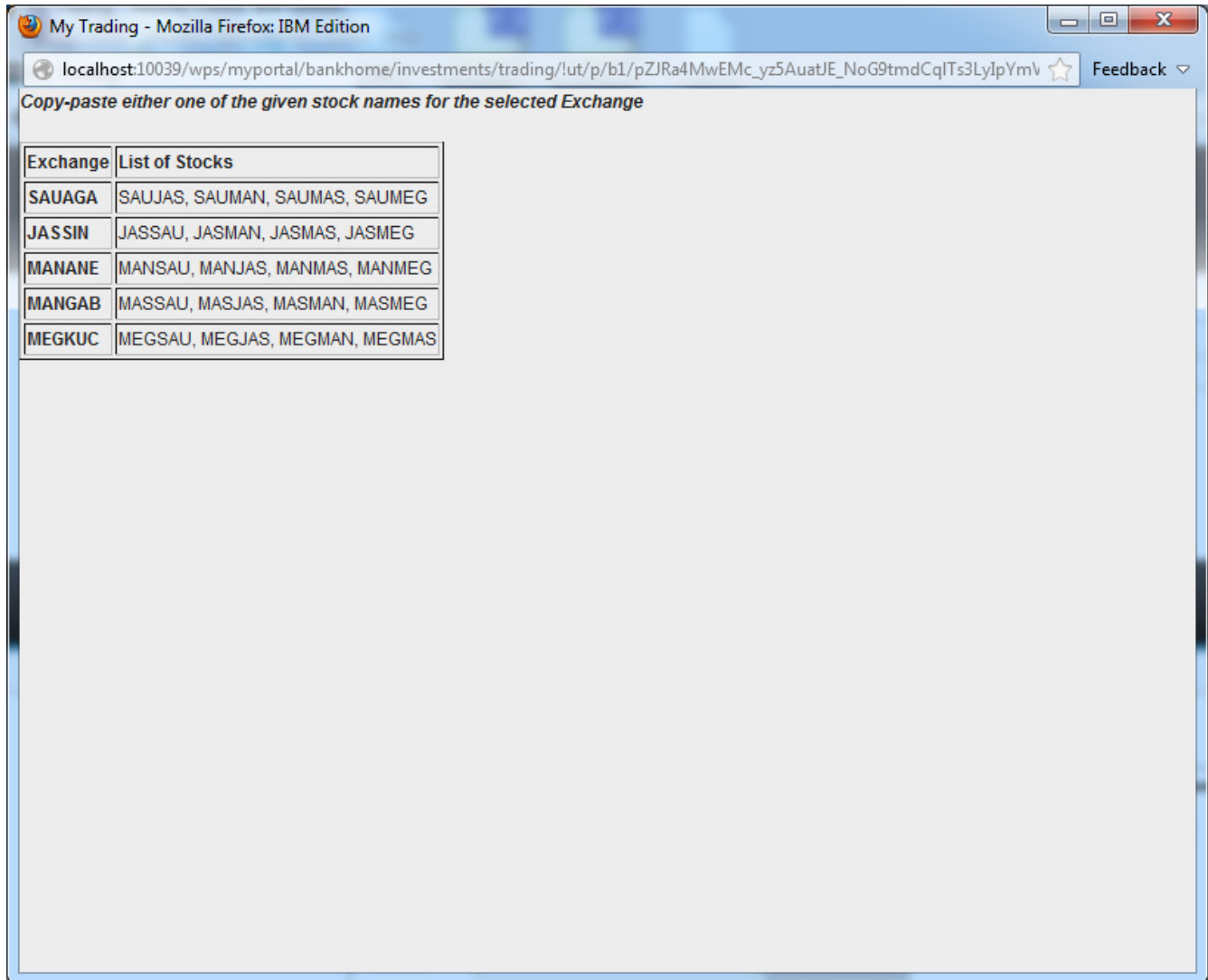
The screenshot shows the 'Investments' section with the same navigation bar as Figure 27. The breadcrumb is 'Investments > My Trading'. The text 'Go to Help mode for the sample input value.' is present. The 'Exchange:' dropdown is set to 'JASSIN' and the 'Stock:' input field contains 'JASSAU'. A 'Get Quotes' button is visible. Below the input fields is a table with the following data:

Name	% Change	Last Price			
JASSAU	+0.56%	512.21↑	Buy	Sell	Watch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

On top of the portlet, a message is displayed "Go to Help mode for the sample input value." Go to **Help mode** to check all the stocks that are available in database as shown in Figure 29:

Figure 29: Help mode of Trading Portlet



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Portfolio Portlet

The Portfolio portlet displays a data table to show the existing stocks in a portfolio. Any stock that you buy, it automatically show up in the portfolio. Similarly, any stock that you sells from the portfolio, is removed as shown in Figure 30:

Figure 30: Portfolio Portlet

Exchange	Name	Buy Price	Current Price	Quantity	Market Value	Profit/Loss		
JASSIN	JASSAU	516.89	512.21	40	20488.4	-4.67 ↓	Buy	Sell

Watchlist Portlet

The Watchlist portlet displays a data table of stocks that are monitored regularly. Any stock that you select from the trading portlet by clicking Watch link, it is automatically added to the watch list. You can buy, sell, and remove stocks from the watch list, as shown in Figure 31:

Figure 31: WatchList Portlet

Exchange	Name	Last Price	% Change			
SAUAGA	SAUJAS	8416.12 ↑	+0.01%	Buy	Sell	Unwatch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Buy and Sell Operations

Buy and Sell operations are performed from the following three portlets:

- Trading Portlet
- Portfolio Portlet
- Watchlist Portlet

Buy and Sell operation procedure is same from all the three portlets.

Buy Operation from Trading Portlet

1. Select an Exchange and enter the stock name, and click Get Quotes. A data table displays information about the stock. Click Buy link next to the stock you want to buy, as shown in Figure 32:

Figure 32: Buy operation in Trading portlet

Investments > My Trading

Go to *Help mode* for the sample input value.

Exchange: MANGAB ▼

Stock: MASSAU

Name	% Change	Last Price			
MASSAU	+0.09%	2415.37 ↑	Buy	Sell	Watch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. A page opens with Exchange name and Stock name. Enter the price and quantity and click Submit, as shown in Figure 33:

Figure 33: Buy stocks page in Trading portlet

Investments > My Trading

Exchange: MANGAB
Stock: MASSAU
Price: 2400
Quantity: 15

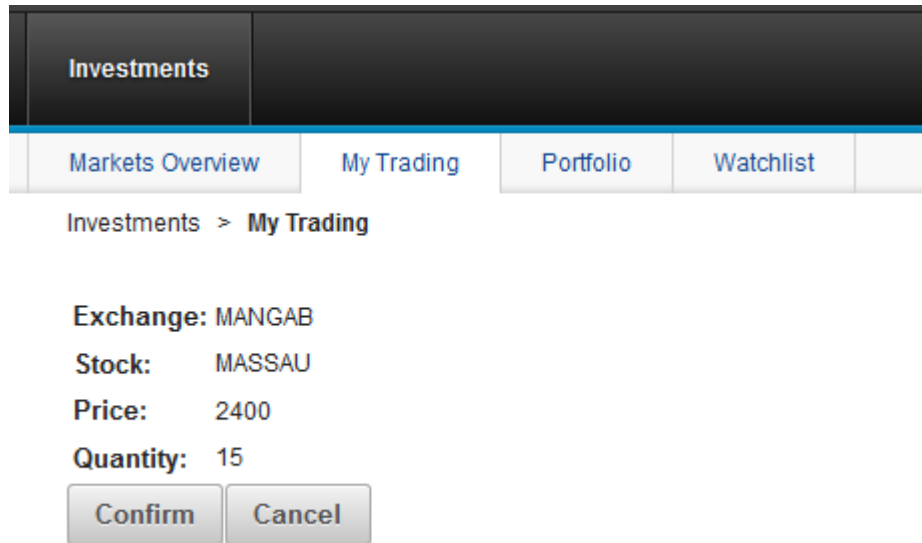
Name	Last Price	% Change
MASSAU	2415.37	+0.09%

Submit Cancel

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

3. A confirmation page displays the details of the transactions. Click Confirm to execute the buy operation as shown in Figure 34:

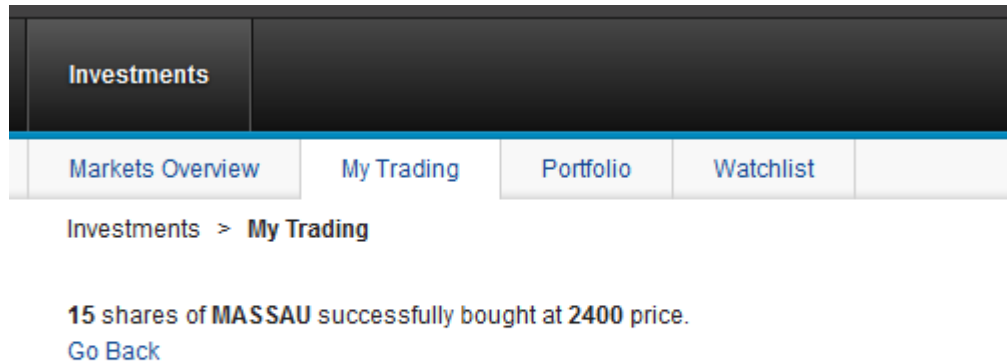
Figure 34: Confirmation page



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

4. A page with a message, for example. “15 shares of **MASSAU** successfully bought at **2400** price” displays, as shown in Figure 35:

Figure 35: Success Page



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Sell Operation from Trading Portlet

1. Select an Exchange enter the stock name, and click Get Quotes. A data table displays information the stock. Click Sell link next to the stock you want to sell as shown in Figure 36:

Figure 36: Sell operation in Trading portlet

The screenshot shows the 'Investments' section of a trading interface. It includes navigation tabs for 'Markets Overview', 'My Trading', 'Portfolio', and 'Watchlist'. Below the tabs, the breadcrumb 'Investments > My Trading' is visible. A text prompt reads 'Go to Help mode for the sample input value.' Below this, there is an 'Exchange:' dropdown menu set to 'SAUAGA' and a 'Stock:' input field containing 'SAUJAS'. A 'Get Quotes' button is positioned to the right of the stock input. Below these elements is a table with the following data:

Name	% Change	Last Price			
SAUJAS	+0.01%	8416.12 ↑	Buy	Sell	Watch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. A page opens with Exchange name and Stock name. Enter the price and quantity and click Submit, as shown in Figure 37:

Figure 37: Stock Sell page in Trading portlet

Investments > My Trading

Exchange: SAUAGA
Stock: SAUJAS
Price: 8000
Quantity: 23

Name	Last Price	% Change
SAUJAS	8416.12	+0.01%

Submit Cancel

3. If selected stock doesn't exist in your Portfolio, an error page is displayed with a message "You cannot sell this stock as it does not exist in your portfolio" as shown in Figure 38:

Figure 38: Sell operation in Trading portlet

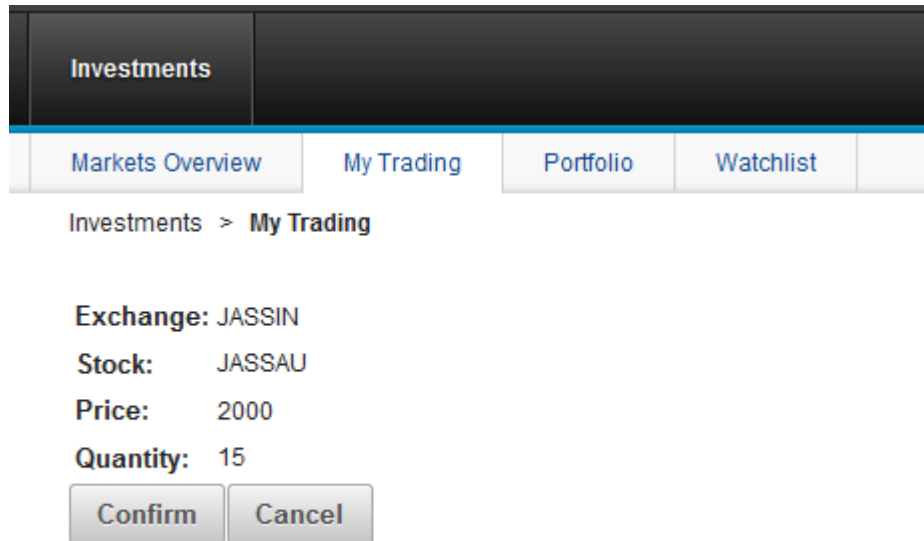
Investments > My Trading

Stock doesn't exist in your portfolio.
[Go Back](#)

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

4. If selected stock exists in your portfolio, a confirmation page displays the details of the transactions. Click Confirm to execute the sell operation as shown in Figure 39:

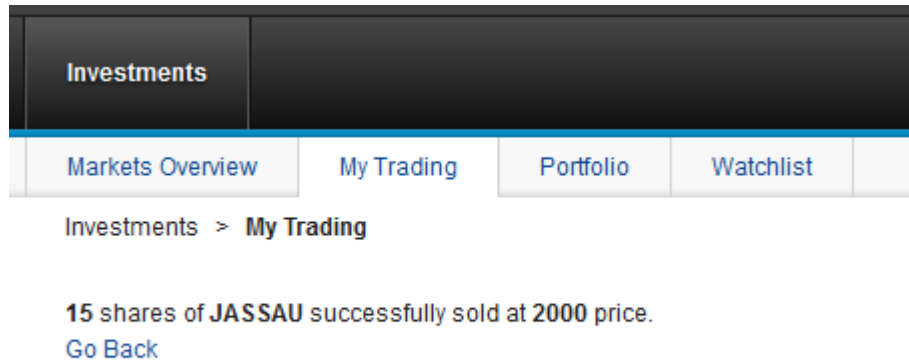
Figure 39: Sell operation confirmation page in Trading portlet



The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

5. A page with a message, for example, "15 shares of JASSAU successfully sold at 2000 price." displays, as shown in Figure 40:

Figure 40: Sell operation in Trading portlet

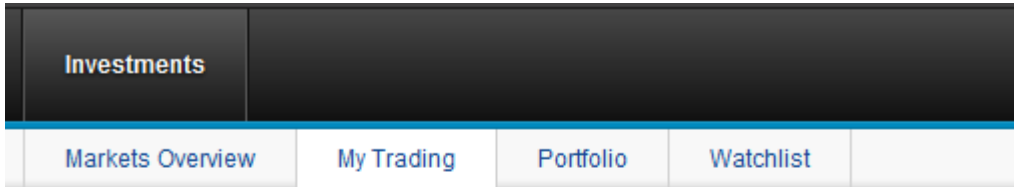


The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Watching operation from Trading portlet

1. Select any Exchange and enter the stock name and click Get Quotes. A data table is displayed with information about that stock as shown in Figure 41:

Figure 41: Watch operation in Trading portlet



Investments > My Trading

Go to Help mode for the sample input value.

Exchange:

Stock:

Name	% Change	Last Price			
MEGSAU	+0.12%	2398.19	Buy	Sell	Watch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. Click Watch to add the selected stock to the Watchlist portlet as shown in Figure 42:

Figure 42: Watch operation in Trading portlet

Exchange	Name	Last Price	% Change			
SAUAGA	SAUJAS	8416.12 ↑	+0.01%	Buy	Sell	Unwatch
MEGKUC	MEGSAU	2398.19 ↑	+0.12%	Buy	Sell	Unwatch

Removing transactions from Watchlist portlet

1. To remove a stock from the Watchlist portlet, click on Unwatch. as shown in Figure 43:

Figure 43: Remove transaction in Trading portlet

Exchange	Name	Last Price	% Change			
SAUAGA	SAUJAS	8416.12 ↑	+0.01%	Buy	Sell	Unwatch
MEGKUC	MEGSAU	2398.19 ↑	+0.12%	Buy	Sell	Unwatch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

2. The stock is removed from the Watchlist portlet as shown in Figure 44:

Figure 44: WatchList portlet

Investments > Watchlist

Exchange	Name	Last Price	% Change			
SAUAGA	SAUJAS	8416.12 ↑	+0.01%	Buy	Sell	Unwatch

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.

Conclusion

In this article you not only saw the power of JSF and portlets working together, but also learnt how Rational Application Developer v8.5 makes it easy to configure and leverage a third party library like Primefaces. This library adds more richness and functionality to basic JSF widgets. JSF portal applications are significant as they bring together best of both worlds.

References

- IBM Webphere Portal Server 8.0
 - <http://www-01.ibm.com/support/docview.wss?uid=swg27007791>
- Rational Application Developer for WebSphere Software 8.5
 - <http://www-01.ibm.com/support/docview.wss?uid=swg24032855>
- Primefaces – JSF component suite.
 - <http://www.primefaces.org>

The sample program is provided to you on an "AS IS" basis, without warranty of any kind. IBM HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow for the exclusion or limitation of implied warranties, so the above limitations or exclusions may not apply to you. IBM shall not be liable for any damages you suffer as a result of using, modifying or distributing the sample program or its derivatives.
