Jakarta Tapestry

Rob Smith
Senior Software Engineer
Object Computing, Inc.
St. Louis, MO

smith_r@ociweb.com

OBJECT COMPUTING, INC.

What is Tapestry?

- An open source, component based framework for developing web applications
- Developed by Howard M. Lewis Ship in 2000
- Makes developing web applications similar to developing traditional GUI applications
- Version 3.0 released in April, 2004
- Allows the ability to write web applications without being concerned with the operation centric Servlet API
- Quoting the Tapestry web page: *"Tapestry reconceptualizes web application development in terms of objects, methods and properties instead of URLs and query parameters"*

Tapestry Architecture

- Built on top of Servlet API
- Requires JDK 1.2 or higher and a Servlet 2.2 (or higher) application server/Servlet container
- Model-View-Controller Architecture

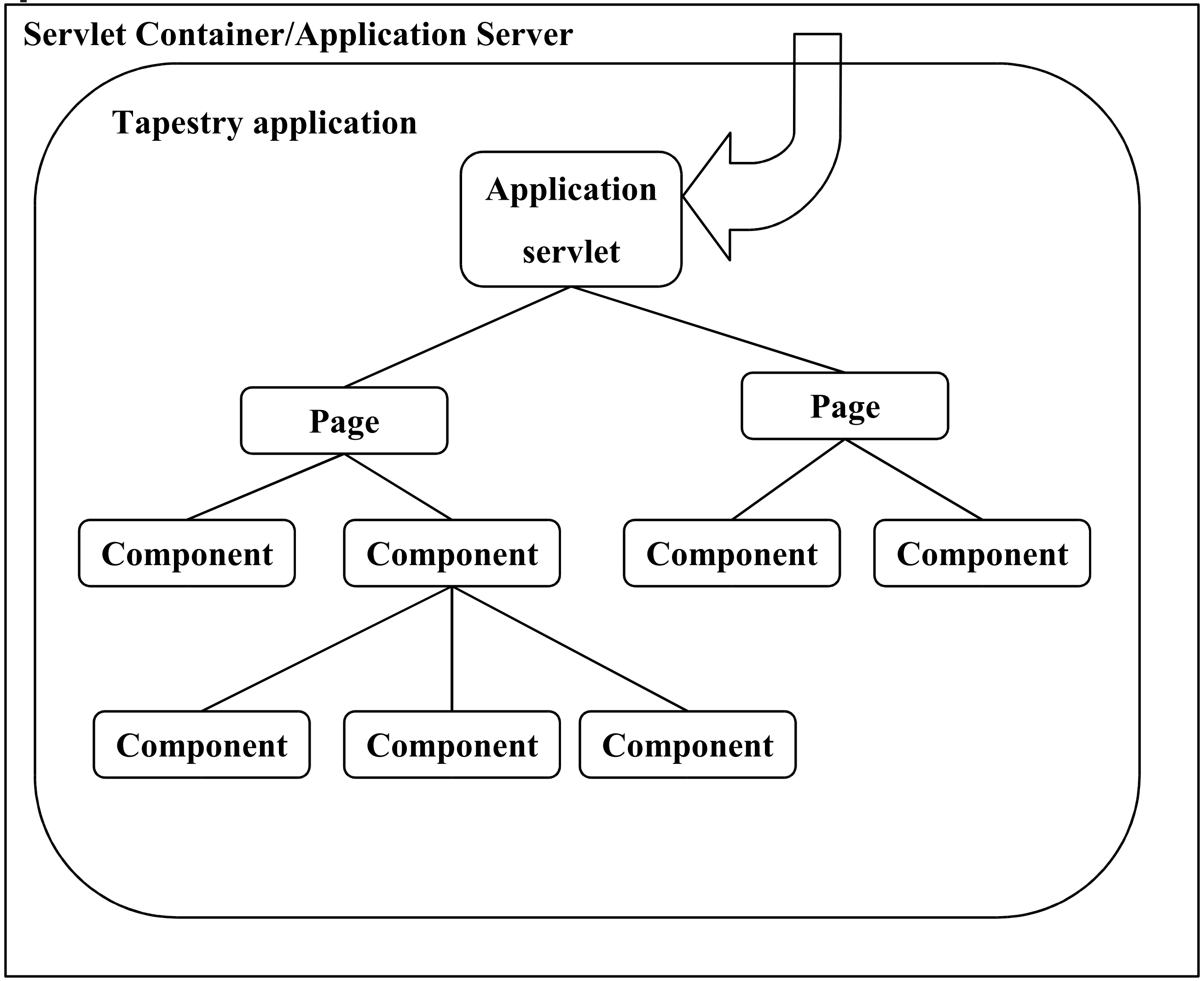
Model-View-Controller

- Model - Made up of Domain Objects
- View - Made up of HTML template
  - Standard HTML with the addition of special markers that define the use of components
  - No Java code ever!
  - Still previews correctly in WYSIWYG editor
  - Located in the web context root directory

Model-View-Controller

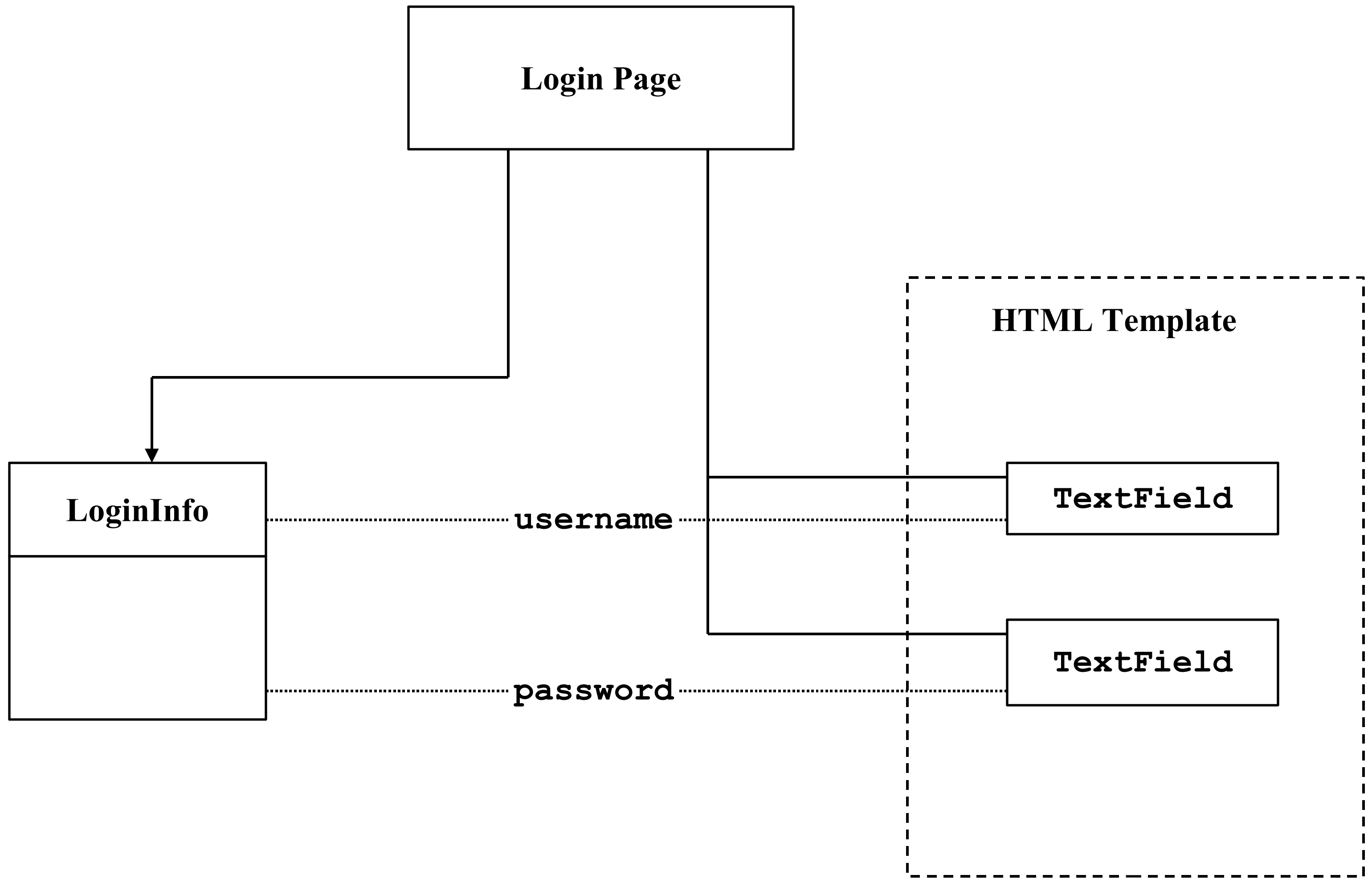
- Controller – Made up of a Page Specification (XML) and Java class
  - Page Specification
    - Ties together placeholders in the HTML template with the rest of the application
    - Typically very short and simple
    - Located in the WEB-INF folder
  - Java Class
    - Class must implement the `org.apache.tapestry.IPage` interface
    - Contains properties (transient and persistent)
    - May contain business logic
    - Located in WEB-INF/classes folder

Tapestry Controller Diagram



Source for diagram: “Tapestry In Action Manning Publications"

Tapestry MVC Diagram



Tapestry Goals

- Simplicity
- Consistency
- Efficiency
- Feedback

Simplicity

- A Tapestry Application contains far less code than a traditional Servlet Application
- Eliminates much of the uninteresting “plumbing” code in traditional Servlet applications
- No need to parse query parameters, manipulate the `HttpSession` object, build URLs, etc
- Allows developers to build applications utilizing stateful Java Beans instead of stateless Servlets
- Allows developers to spend their time writing application-specific logic

Consistency

- Provides a consistent approach to developing pages for a web application
- Helps eliminate the guesswork that occurs in traditional Servlet applications
- Pages in a Tapestry application all work similarly because they are built with the same reusable components

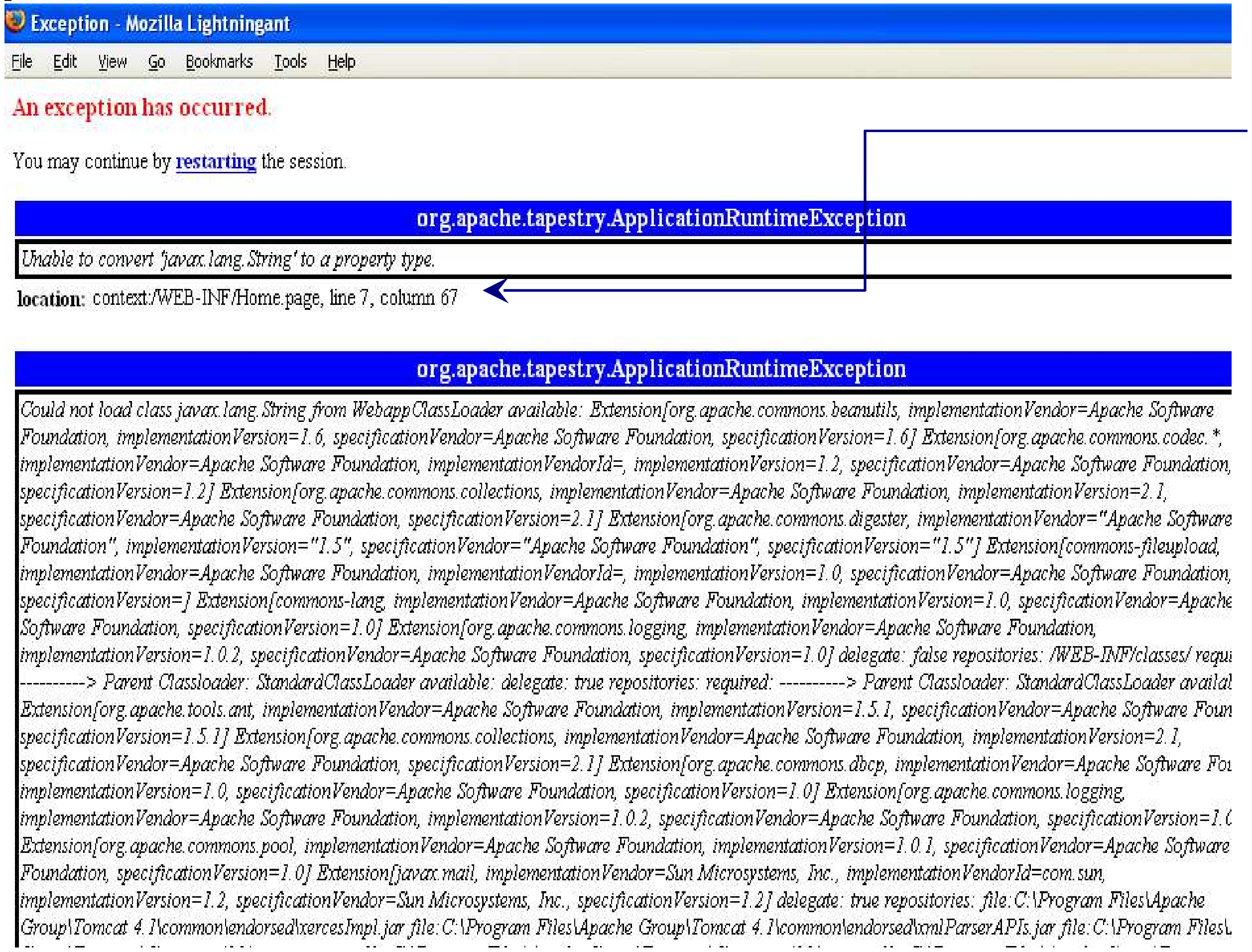
Efficiency

- Tapestry applications are highly scalable and Tapestry is implemented utilizing caches and object pools to minimize processing time for each request
- Tapestry applications have similar performance to traditional Servlet applications
- Developer efficiency by increasing developer productivity

Feedback

- Line precise error reporting
- No more looking through stack traces to determine a config file has an error in it
- Uncaught exceptions get report containing entire stack of exceptions and each exception's properties
  - Dumps out all the Servlet API objects (Servlet, HttpServletRequest, HttpSession, ServletContext)
  - Dumps out all JVM System Properties
- Framework provides a lot of feedback on errors without having to use the debugger

Line Precise Error Reporting


Exception - Mozilla Lightningant
File Edit View Go Bookmarks Tools Help

An exception has occurred.

You may continue by restarting the session.

org.apache.tapestry.ApplicationRuntimeException

Unable to convert 'javax.lang.String' to a property type.

location: context:/WEB-INF/Home.page, line 7, column 67

org.apache.tapestry.ApplicationRuntimeException

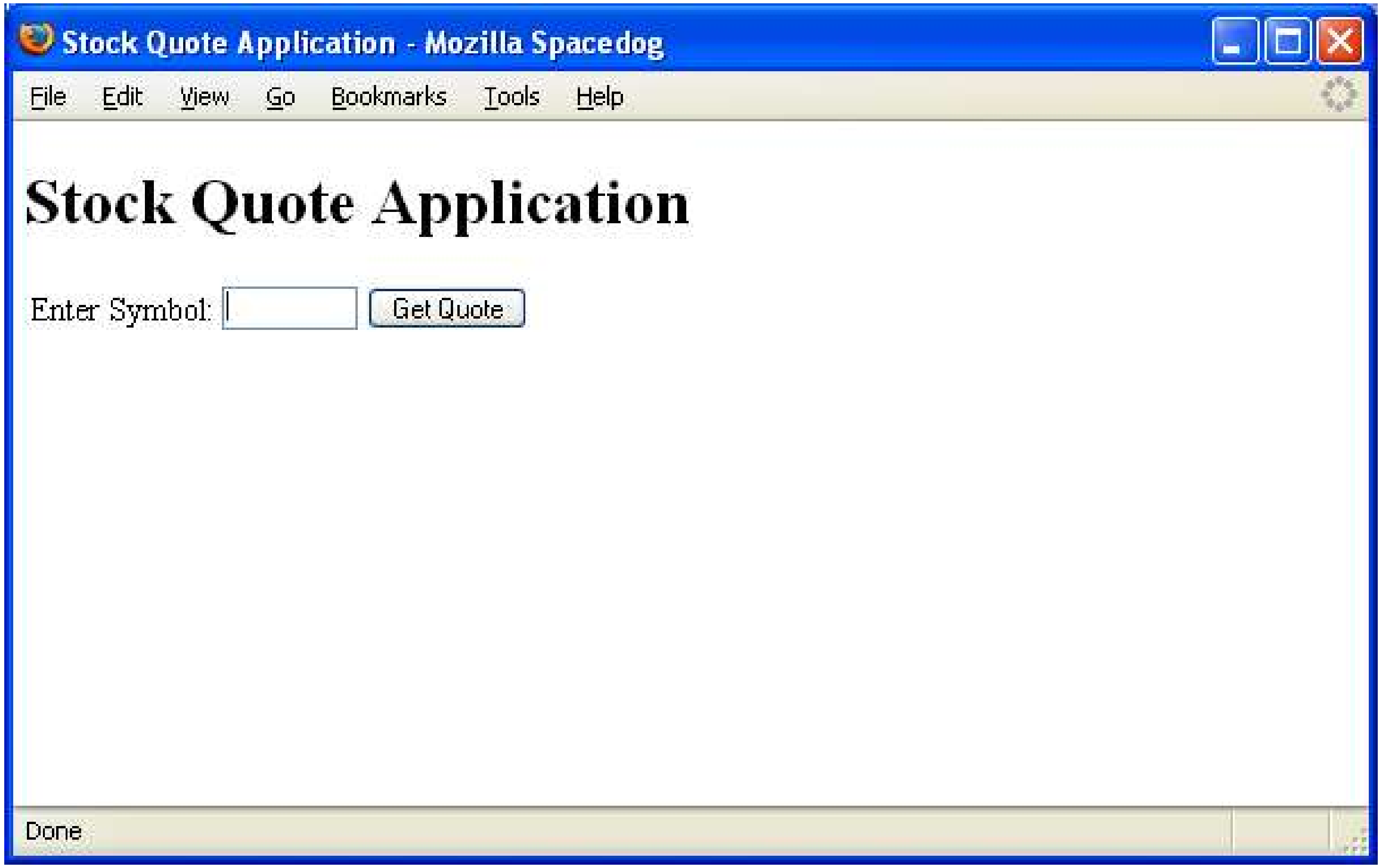
```
Could not load class javax.lang.String from WebappClassLoader available: Extension[org.apache.commons.beanutils, implementationVendor=Apache Software
Foundation, implementationVersion=1.6, specificationVendor=Apache Software Foundation, specificationVersion=1.6] Extension[org.apache.commons.codec.*,
implementationVendor=Apache Software Foundation, implementationVendorId=, implementationVersion=1.2, specificationVendor=Apache Software Foundation,
specificationVersion=1.2] Extension[org.apache.commons.collections, implementationVendor=Apache Software Foundation, implementationVersion=2.1,
specificationVendor=Apache Software Foundation, specificationVersion=2.1] Extension[org.apache.commons.digester, implementationVendor="Apache Software
Foundation", implementationVersion="1.5", specificationVendor="Apache Software Foundation", specificationVersion="1.5"] Extension[commons-fileupload,
implementationVendor=Apache Software Foundation, implementationVendorId=, implementationVersion=1.0, specificationVendor=Apache Software Foundation,
specificationVersion=] Extension[commons-lang, implementationVendor=Apache Software Foundation, implementationVersion=1.0, specificationVendor=Apache
Software Foundation, specificationVersion=1.0] Extension[org.apache.commons.logging, implementationVendor=Apache Software Foundation,
implementationVersion=1.0.2, specificationVendor=Apache Software Foundation, specificationVersion=1.0] delegate: false repositories: /WEB-INF/classes/ requ
----------> Parent Classloader: StandardClassLoader available: delegate: true repositories: required: ----------> Parent Classloader: StandardClassLoader availal
Extension[org.apache.tools.ant, implementationVendor=Apache Software Foundation, implementationVersion=1.5.1, specificationVendor=Apache Software Foun
specificationVersion=1.5.1] Extension[org.apache.commons.collections, implementationVendor=Apache Software Foundation, implementationVersion=2.1,
specificationVendor=Apache Software Foundation, specificationVersion=2.1] Extension[org.apache.commons.dbcp, implementationVendor=Apache Software Fo
implementationVersion=1.0, specificationVendor=Apache Software Foundation, specificationVersion=1.0] Extension[org.apache.commons.logging,
implementationVendor=Apache Software Foundation, implementationVersion=1.0.2, specificationVendor=Apache Software Foundation, specificationVersion=1.(
Extension[org.apache.commons.pool, implementationVendor=Apache Software Foundation, implementationVersion=1.0.1, specificationVendor=Apache Software
Foundation, specificationVersion=1.0] Extension[javax.mail, implementationVendor=Sun Microsystems, Inc., implementationVendorId=com.sun,
implementationVersion=1.2, specificationVendor=Sun Microsystems, Inc., specificationVersion=1.2] delegate: true repositories: file:C:\Program Files\Apache
Group\Tomcat 4.1\common\endorsed\xercesImpl.jar file:C:\Program Files\Apache Group\Tomcat 4.1\common\endorsed\xmlParserAPIs.jar file:C:\Program Files\
```

Shows the file and line number that caused the error

Sample Application

- Simple stock quote application
- Stock symbol entry field
- Error message on failure
- Uses basic Tapestry components

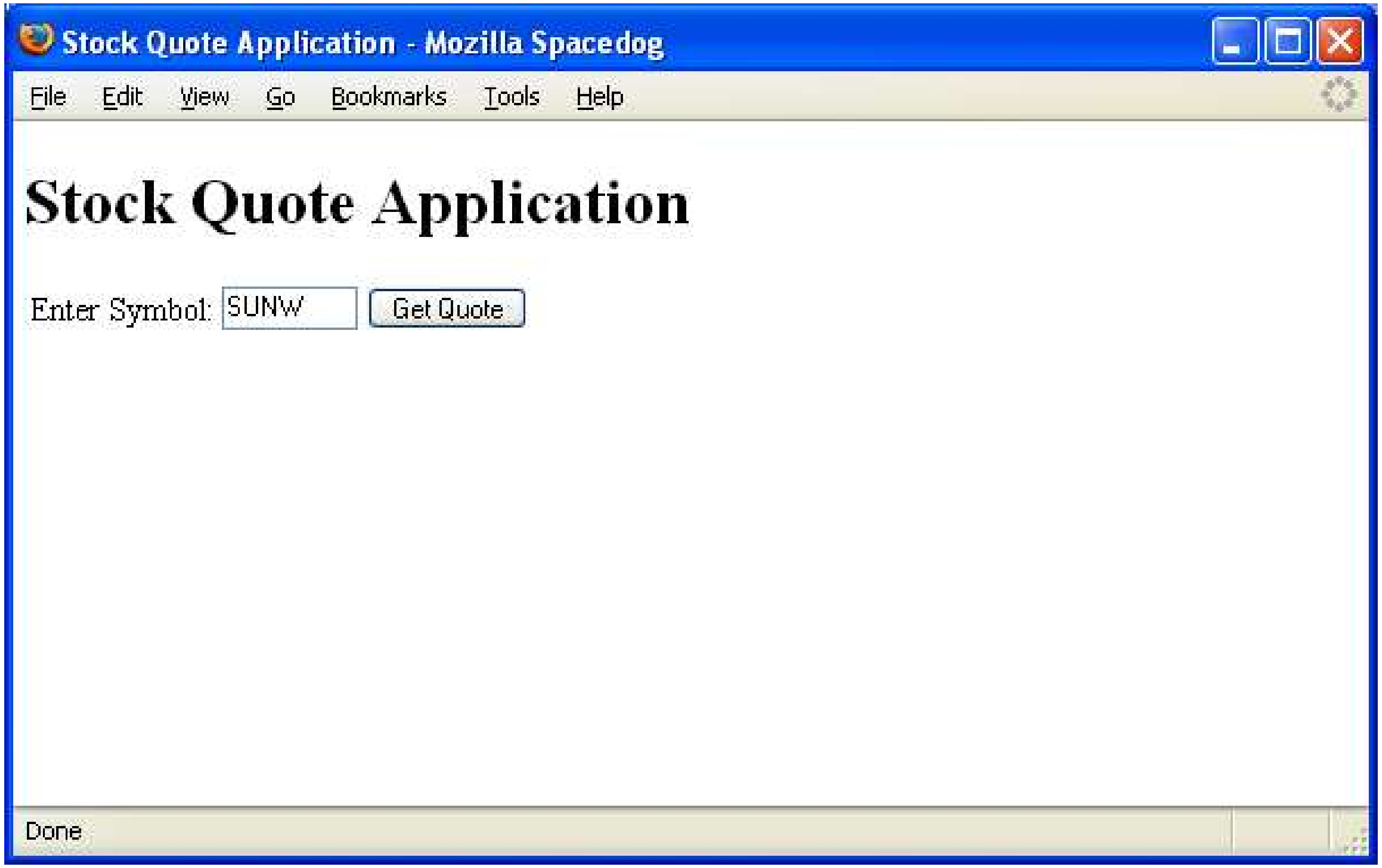
Sample Application


Stock Quote Application - Mozilla Spacedog
File Edit View Go Bookmarks Tools Help
Get Quote
Done

Stock Quote Application

Enter Symbol:

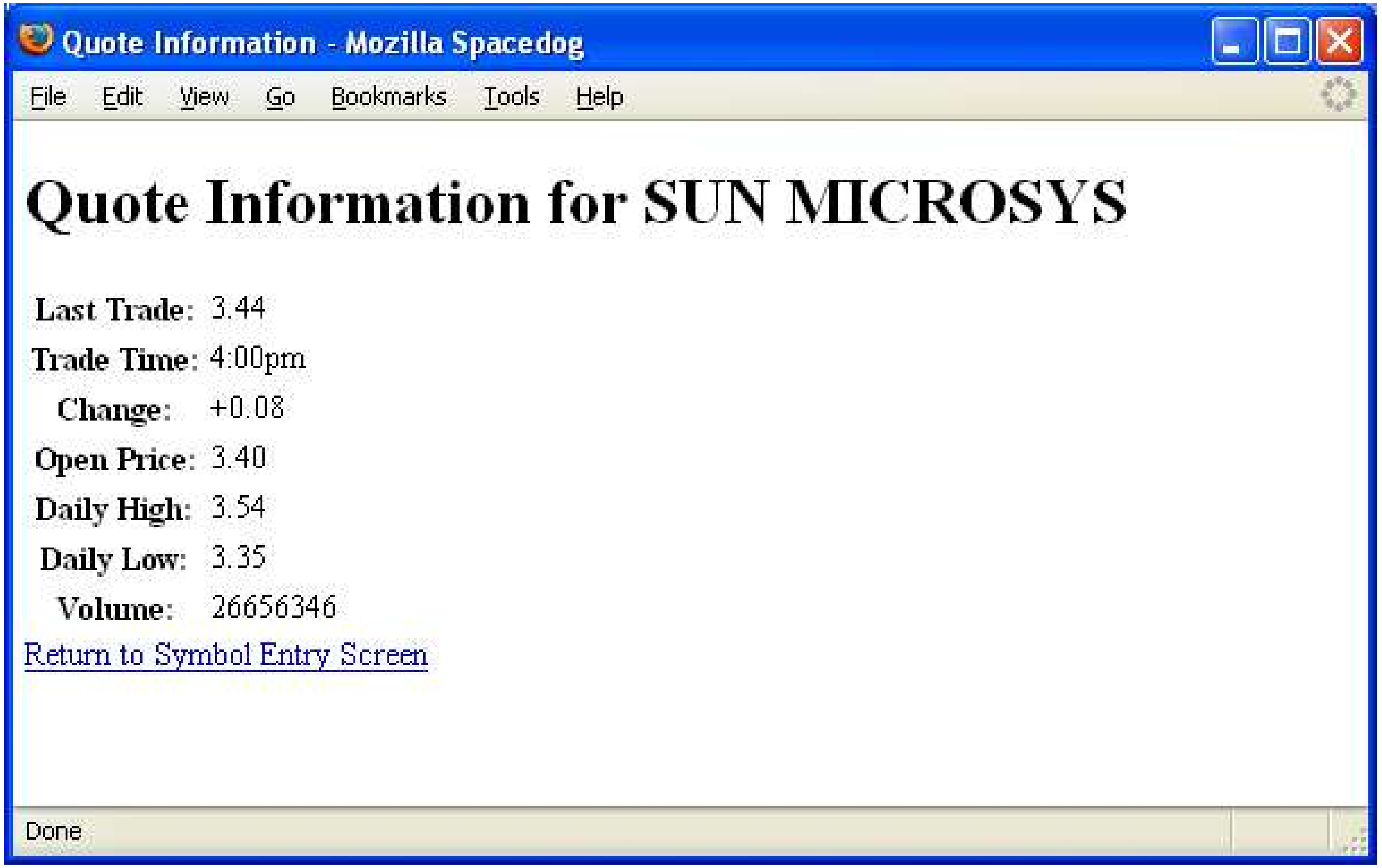
Sample Application (cont.)


Stock Quote Application - Mozilla Spacedog
File Edit View Go Bookmarks Tools Help
Get Quote
Done

Stock Quote Application

Enter Symbol: SUNW

Sample Application (cont.)


Quote Information - Mozilla Spacedog
File Edit View Go Bookmarks Tools Help

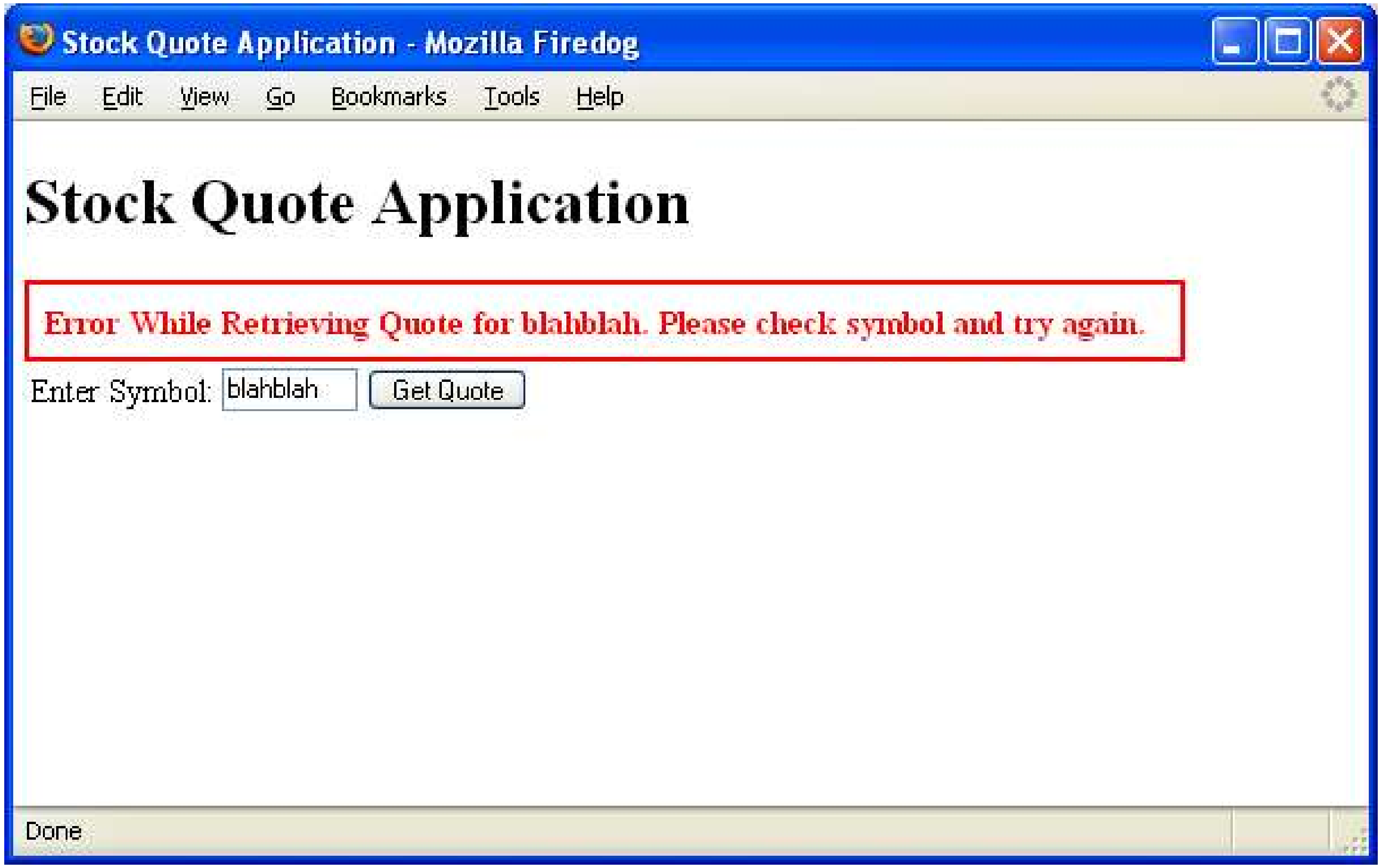
Quote Information for SUN MICROSYS

Last Trade:	3.44
Trade Time:	4:00pm
Change:	+0.08
Open Price:	3.40
Daily High:	3.54
Daily Low:	3.35
Volume:	26656346

Return to Symbol Entry Screen

Done

Sample Application (cont.)


Stock Quote Application - Mozilla Firedog
File Edit View Go Bookmarks Tools Help

Stock Quote Application

Error While Retrieving Quote for blahblah. Please check symbol and try again.

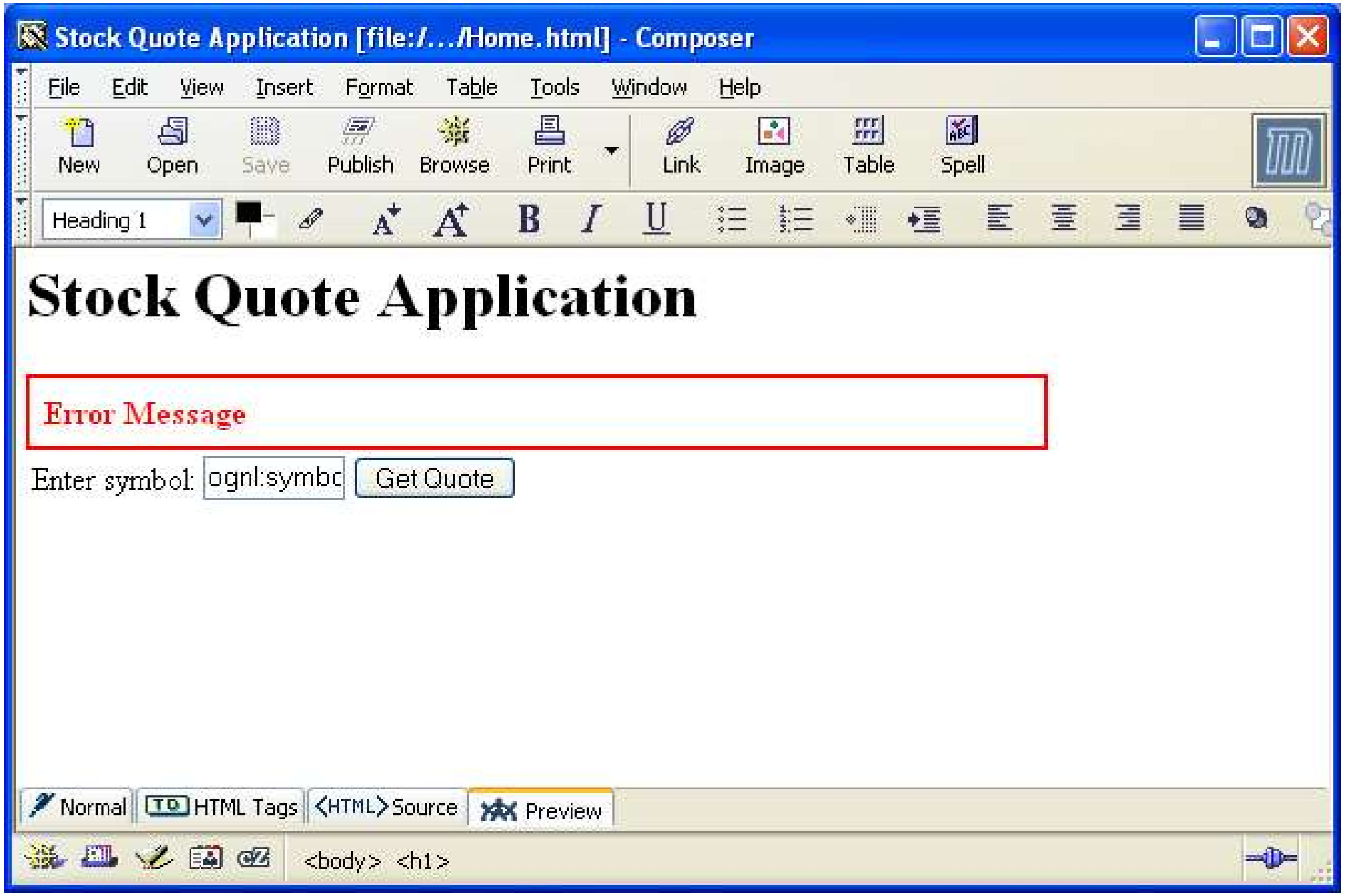
Enter Symbol: blahblah Get Quote

Done

HTML Template

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                    Error Message</span>
            </td></tr>
        </table></div>
    </div>
    <form jwcid="@Form" listener="ognl:listeners.formSubmit">
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

HTML Template WYSIWYG Preview


Stock Quote Application [file:/.../Home.html] - Composer
File Edit View Insert Format Table Tools Window Help
New Open Save Publish Browse Print Link Image Table Spell
Heading 1
Normal HTML Tags Source Preview
<body> <h1>

Stock Quote Application

Error Message

Enter symbol: ognl:symbo Get Quote

HTML Template (cont.)

jwcid – Java Web Component Id

Identifies the component to be used in the template

```
<html><head>
    <link rel="stylesheet" type="tex
    <title>Stock Quote Application</
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                    Error Message</span>
            </td></tr>
        </table></div>
    </div>
    <form jwcid="@Form" listener="ognl:listeners.formSubmit">
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

HTML Template (cont.)

@Type - Defines the ***component*** type

These are anonymous component definitions (Tapestry will assign them a unique Id)

These are all built in components

```
<html><head>
    <link rel="stylesheet" type=
    <title>Stock Quote Applicati
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                    Error Message</span>
            </td></tr>
        </table></div>
    </div>
    <form jwcid="@Form" listener="ognl:listeners.formSubmit">
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

HTML Template (cont.)

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                    Error Message</span>
            </td></tr>
        </table></div>
    </div>
    <form jwcid="@Form" listener="ognl:listeners.formSubmit">
```

more on OGNL later …

condition is a ***parameter*** of the Conditional component

It is specified with an Object Graph Navigation Language (OGNL) expression

It means if the error property of our page is not null and not empty to display it

The body of the ***Conditional*** component is discarded at runtime if the condition parameter evaluates to false

HTML Template (cont.)

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                   Error Message</span>
          </td></tr>
        </tab
    </div>
    <form jwc
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

The ***Insert*** component outputs text in generated page

The ***value*** parameter specifies the text to insert

HTML Template (cont.)

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><
    <h1>St
    <div j
      <div
        <t
                                                                         ">

         </td></tr>
        </table></div>
    </div>
    <form jwcid="@Form" listener="ognl:listeners.formSubmit">
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

The ***Form*** component outputs a HTML form element

The ***listener*** parameter specifies the method to invoke on the Page class

Notice that we do not have to worry about the action URL, Tapestry handles this for us

HTML Template (cont.)

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
                                                             r">


        <
    </div
    <form
      <table><tr>
          <td>Enter symbol:</td>
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

The ***TextField*** component outputs a HTML input text field that is used to edit a String property

The ***value*** parameter is the page property that is read on the initial render and updated when the form is submitted

HTML Template (cont.)

```
<html><head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head><body>
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error">
        <table><tr><td>
              <span class="error" jwcid="@Insert" value="ognl:error">
                   Error Message</span>
            </td></tr>
        </table></div>
    </di
    <for
      <ta
          <td><input jwcid="@TextField" value="ognl:symbol" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr></table>
    </form></body></html>
```

The ***Submit*** component outputs a HTML submit button

The ***value*** parameter will be rendered as the submit button's text

HTML Template Notes

- Component tags must be well formed
- All other template content is free-form
- HTML templates should be located in the context root directory

Object Graph Navigation Language

- The ognl: prefix used throughout the HTML template and in the page specification (coming next) refers to the Object Graph Navigation Language
- OGNL is a powerful open source expression language used to get and set properties of Java objects
- Beyond getting and setting simple properties of Java objects, OGNL expressions can do almost anything you can do with Java expressions: invoke methods, perform comparisons, and much more
- For more information visit the OGNL homepage at http://www.ognl.org

Page Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.Home">
  <property-specification name="error" type="java.lang.String"/>
  <property-specification name="symbol" type="java.lang.String"/>
  <context-asset name="stylesheet" path="/css/style.css"/>
</page-specification>
```

Page Specification (cont.)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Ap                                                      /EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.Home">
  <property-specification name="error" type="java.lang.String"/>
  <property-specification name="symbol" type="java.lang.String"/>
</page-specification>
```

The Java class that Tapestry will instantiate for the page

Page Specification (cont.)

Creates Java Bean properties for the page

Tapestry will create a subclass with the specified properties at runtime

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTY
  "-//Ap                                                    /EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.Home">
  <property-specification name="error" type="java.lang.String"/>
  <property-specification name="symbol" type="java.lang.String"/>
</page-specification>
```

Page Class

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(String error);

    public abstract String getSymbol();
    public abstract void setSymbol(String symbol);

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
        }
        QuoteService quoteService = …
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
```

Page Class (cont.)

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(String error);

    public a
    public abstract void setSymbol(String symbol);

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
        }
        QuoteService quoteService = …
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
```

org.apache.tapestry.html.BasePage is the class to subclass for HTML pages

Page Class (cont.)

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(String error);

    public abstract String getSymbol();
    public abstract void setSymbol(String symbol);
```

The page class is abstract with abstract Java bean methods

The error and symbol property were specified in the page specification

Tapestry will create a subclass at runtime with the appropriate properties

```
        QuoteService quoteService = …
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
}
```

Page Class (cont.)

Listener method invoked when the form is submitted

The method is invoked after the page properties have been set

```
public abstract class Home extend
    public abstract String getErr
    public abstract void setError

    public abstract String getSym
    public abstract void setSymbol(String symbol);

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
        }
        QuoteService quoteService = …
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
```

Page Class (cont.)

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(St

    public abstract String getSymbol();
    public abstract void setSymbol(String symbol);

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
        }
        QuoteService quoteService = …
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
}
```

Notice the use of the abstract methods in the listener method

Page Class (cont.)

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(String error);

    public abstract String getSymbol();
    public abstract voi

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
        }
        QuoteService q
        try {
            Quote quote
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
```

If the user does not type a valid symbol …

or a quote is not available for the symbol entered, the error property is set and the current page stays active and renders the response

Page Class (cont.)

```
public abstract class Home extends BasePage {
    public abstract String getError();
    public abstract void setError(String error);

    public abstract String getSymbol();
    public abstract void setSymbol(String symbol);

    public void formSubmit(IRequestCycle cycle) {
        String symbol = getSymbol();
        if (symbol == null || symbol.trim().length() == 0) {
            setError("Must type in a Symbol");
            return;
```

If we retrieve a quote for the symbol entered the ShowQuote page is activated and has its Quote property set.

```
        try {
            Quote quote = quoteService.retrieveQuote(symbol);
            ShowQuote showQuotePage = (ShowQuote) cycle.getPage("ShowQuote");
            showQuotePage.setQuote(quote);
            cycle.activate(showQuotePage);
        } catch (QuoteFetchException e) {
            setError("Error While Retrieving Quote for " + symbol +
                     ". Please check symbol and try again.");
        }
    }
}
```

ShowQuote HTML Template

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Quote Information</title>
  </head>
  <body>
    <h1>Quote Information for
      <span jwcid="@Insert" value="ognl:quote.company">Yahoo</span>
    </h1>
    <table>
      <tr><th>Last Trade:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.value">25.00</span></td></tr>
      <tr><th>Trade Time:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.time">3:30pm</span></td></tr>
      <tr><th> Change:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.net">+1.50</span></td></tr>
      <tr><th> Open Price:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.openPrice">23.50</span></td></tr>
```

ShowQuote HTML Template (cont.)

```
      <tr><th> Daily High:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.dailyHigh">25.50</span></td></tr>
      <tr><th> Daily Low:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.dailyLow">24.50</span></td></tr>
      <tr><th> Volume:</th>
        <td><span jwcid="@Insert"
                  value="ognl:quote.volume">24.50</span></td></tr>
    </table>
    <span jwcid="@PageLink" page="Home">Return to Symbol Entry Screen</span>
  </body>
</html>
```

The ***PageLink*** component creates a hyperlink to the page specified by the page ***parameter***

ShowQuote Page Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.ShowQuote">
    <property-specification name="quote" type="com.ociweb.quote.Quote"/>
</page-specification>
```

ShowQuote Page Class

```
public abstract class ShowQuote extends BasePage {
    public abstract Quote getQuote();
    public abstract void setQuote(Quote quote);
}
```

The only reason this class was created was to be able to call the setQuote method from the Home page.

The alternative approach is demonstrated on the next slide.

No Page Class for ShowQuote

Page Specification:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="org.apache.tapestry.html.BasePage">
    <property-specification name="quote" type="com.ociweb.quote.Quote"/>
</page-specification>
```

Home Page class:

```
public void formSubmit(IRequestCycle cycle) {
    …
    try {
        Quote quote = …
        IPage showQuotePage = cycle.getPage("ShowQuote");
        showQuotePage.setProperty("quote", quote);
        cycle.activate(showQuotePage);
    } catch (QuoteFetchException e) {
        …
    }
}
```

web.xml

```
<servlet>
  <servlet-name>quote</servlet-name>
  <servlet-class>org.apache.tapestry.ApplicationServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>quote</servlet-name>
  <url-pattern>/app</url-pattern>
</servlet-mapping>
```

- All requests are handled by the ApplicationServlet class provided by Tapestry
- Typical convention is to map all requests to /app but is not required

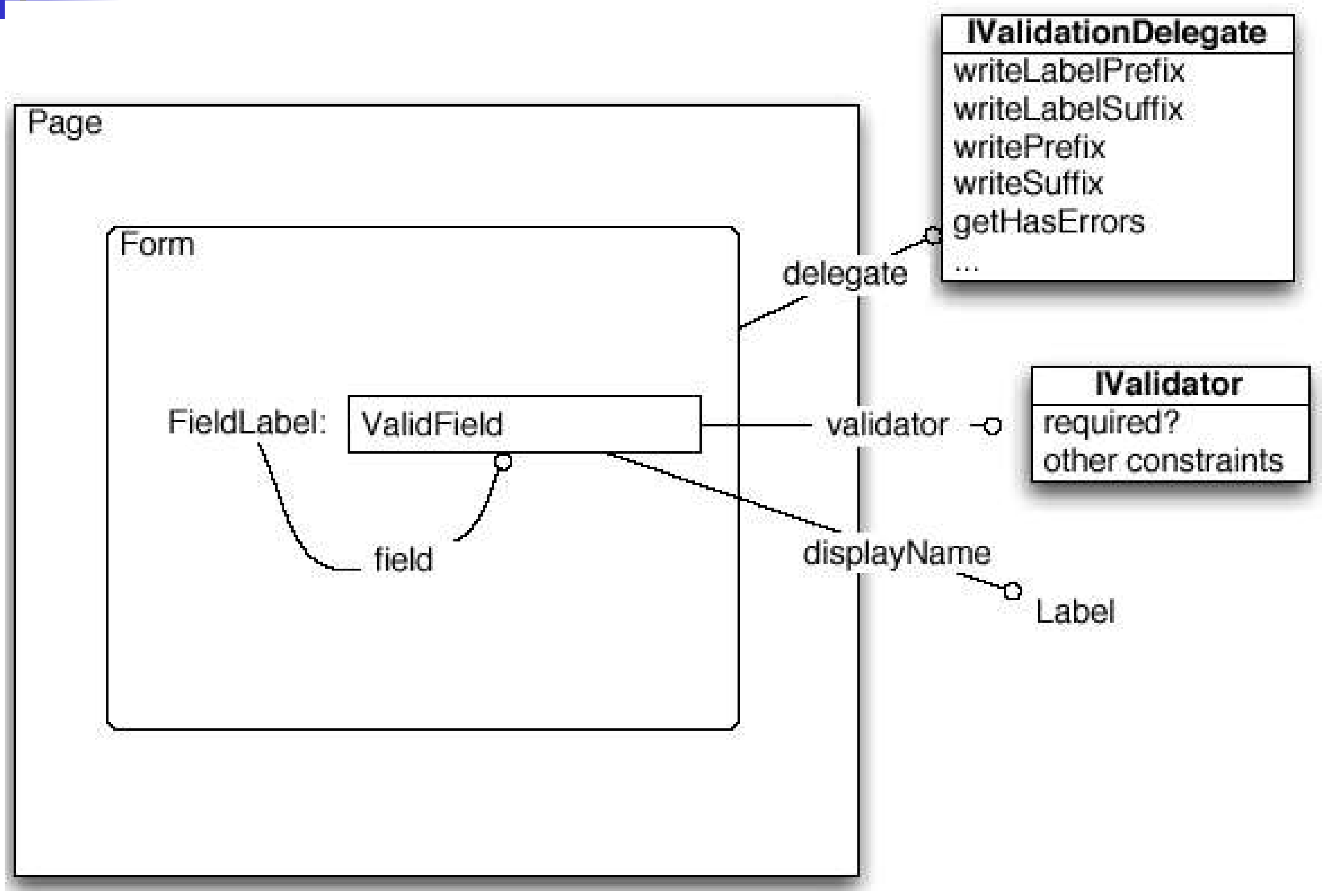
Form Input Validation

- Tapestry provides components for validating user input
- The validation subsystem is centered on the *ValidField* component
- The *ValidField* component is a variation of the *TextField* component that has the ability to edit not just String properties but also Numbers, Dates, etc.
- The *ValidField* component helps provide useful feedback to the user in terms of client-side validation and visual indications of errors in the form
- The *FieldLabel* component can be attached to a *ValidField* component to display itself differently if the field is in error

Form Input Validation (cont.)

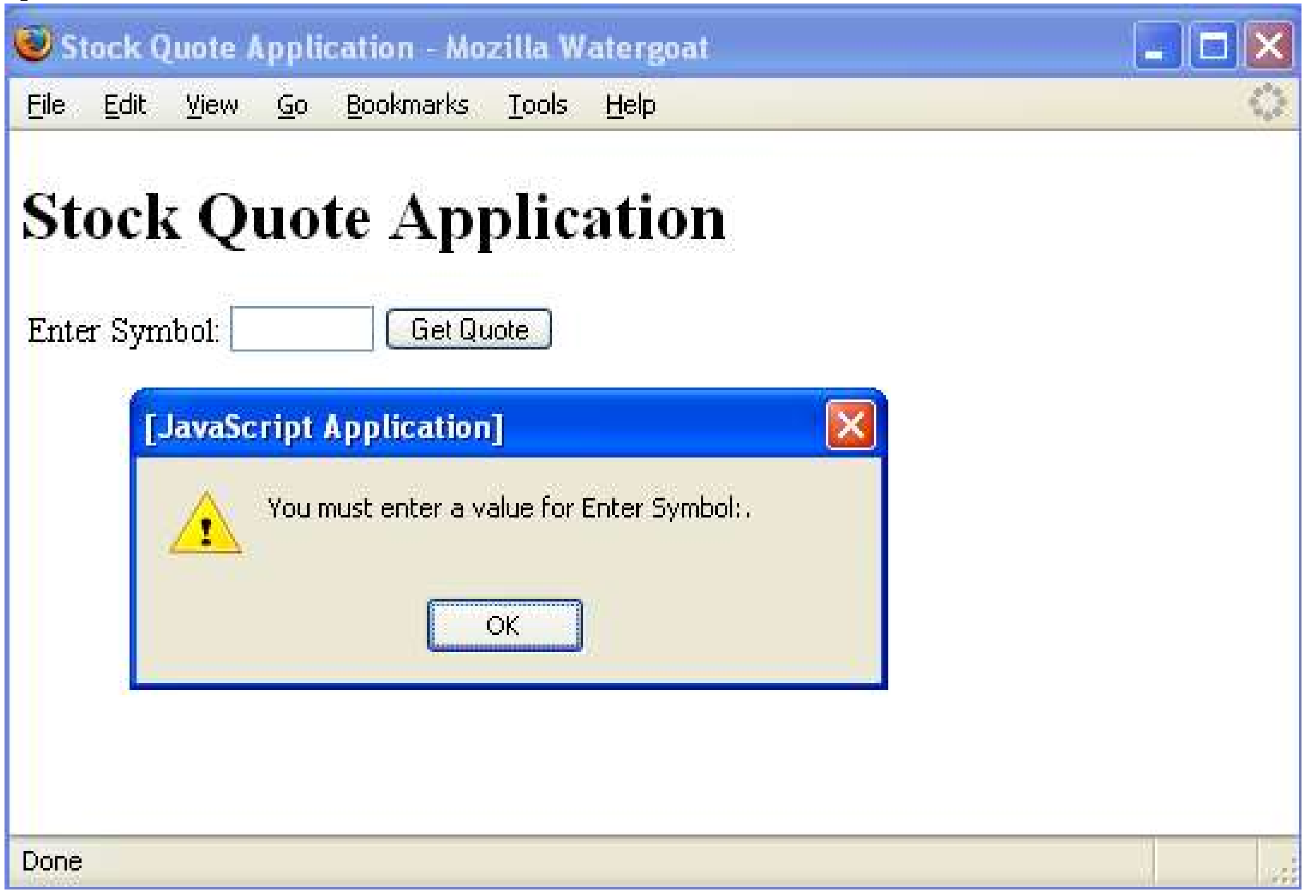
- The `org.apache.tapestry.valid.IValidationDelegate` interface is responsible for tracking validation errors in a form.
- The `org.apache.tapestry.valid.IValidator` interface is responsible for processing a component’s input and validating the value

Validation Visualization



Thanks to Erik Hatcher http://www.ehatchersolutions.com/ for allowing me to use this diagram

Validation Demo


Stock Quote Application - Mozilla Watergoat

Stock Quote Application

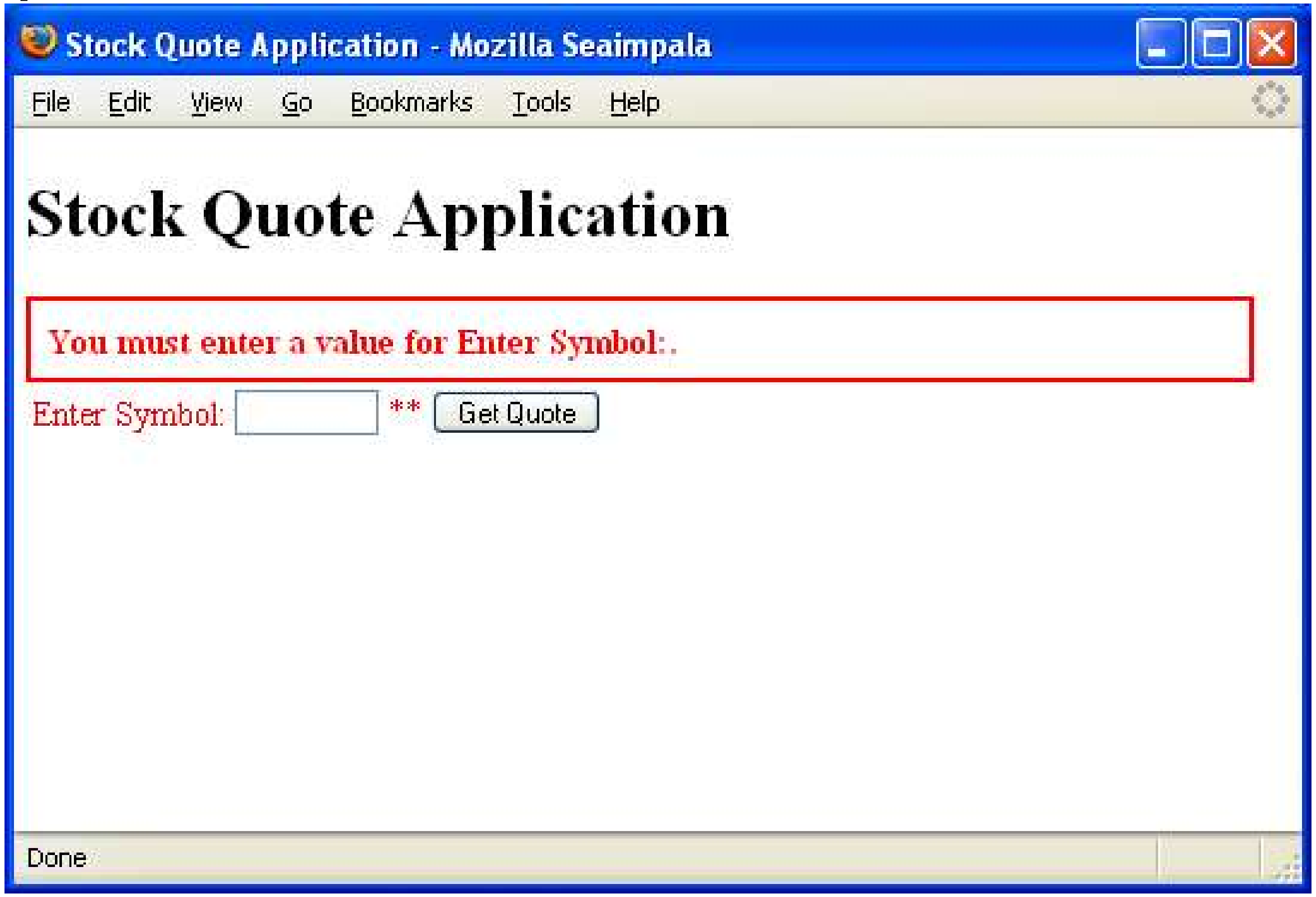
Enter Symbol: [ ] Get Quote

[JavaScript Application]

You must enter a value for Enter Symbol:.

Validation on the client side

Validation Demo


Stock Quote Application - Mozilla Seaimpala
File Edit View Go Bookmarks Tools Help

Stock Quote Application

You must enter a value for Enter Symbol:.

Enter Symbol: ** Get Quote

Validation on the server side

HTML Template

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head>
  <body jwcid="@Body">
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error"><table><tr><td>
              <span jwcid="@Insert" value="ognl:error">
                    Error Message
              </span></td></tr></table></div>
    </div>
    <div jwcid="@Conditional" condition="ognl:beans.delegate.hasErrors">
      <div class="error">
        <table><tr><td><span class="error">
              <span class="error" jwcid="@Delegator"
                    delegate="ognl:beans.delegate.firstError">
                    Validation Error Message
              </span></span></td></tr></table></div>
    </div>
```

HTML Template (cont.)

```
<form jwcid="@Form" delegate="ognl:beans.delegate"
      listener="ognl:listeners.formSubmit">
      <table>
        <tr>
          <td><span jwcid="@FieldLabel"
                    field="ognl:components.inputSymbol">Enter
   symbol:</span></td>
          <td><input jwcid="inputSymbol" type="text" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

HTML Template

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head>
  <body jwcid="@Body">
    <h1>Stock Quote Application</h1>
    <div jwcid="@Conditional" condition="ognl:error">
      <div class="error"><table><tr><td>
              <span jwcid="@Insert" value="ognl:error">
```

The ***Body*** component renders the html body element and is needed for JavaScript that is generated for client-side validation

```
    </d
    <div jwcid="@Conditional" condition="ognl:beans.delegate.hasErrors">
      <div class="error">
        <table><tr><td><span class="error">
              <span class="error" jwcid="@Delegator"
                    delegate="ognl:beans.delegate.firstError">
                    Validation Error Message
              </span></span></td></tr></table></div>
    </div>
```

HTML Template

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <title>Stock Quote Application</title>
  </head>
  <body jwc
    <h1>Sto
    <div jw
      <div
          <span jwcid="@Insert" value="ognl:error">
                Error Message
          </span></td></tr></table></div>
    </div>
    <div jwcid="@Conditional" condition="ognl:beans.delegate.hasErrors">
      <div class="error">
        <table><tr><td><span class="error">
          <span class="error" jwcid="@Delegator"
                delegate="ognl:beans.delegate.firstError">
                Validation Error Message
          </span></span></td></tr></table></div>
    </div>
```

The ***Conditional*** and ***Delegator*** components are used to display the first error reported by the ***org.apache.tapestry.valid.IValidationDelegate*** implementation defined in the page specification

HTML Template (cont.)

```
<form jwcid="@Form" delegate="ognl:beans.delegate"
      listener="ognl:listeners.formSubmit">
      <table>
        <tr>
          <td><s                                        onents.inputSymbol">
              Enter symbol:</span></td>
          <td><input jwcid="inputSymbol" type="text" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Setting the ***delegate*** parameter on the ***Form*** component enables form validation

HTML Template (cont.)

```
<form jwcid="@Form" delegate="ognl:beans.delegate"
      listener="ognl:listeners.formSubmit">
      <table>
        <tr>
          <td><span jwcid="@FieldLabel" field="ognl:components.inputSymbol">
              Enter symbol:</span></td>
          <td><input jwcid="inputSymbol" type="text" size="8"/></td>
          <                                                  e"/></td>
        </t
      </tab
    </form>
  </body>
</html>
```

The ***FieldLabel*** component is attached to the inputSymbol component to alter its view if an invalid value is entered

HTML Template (cont.)

```
<form jwcid="@Form" delegate="ognl:beans.delegate"
      listener="ognl:listeners.formSubmit">
      <table>
        <tr>
          <td><span jwcid="@FieldLabel" field="ognl:components.inputSymbol">
              Enter symbol:</span></td>
          <td><input jwcid="inputSymbol" type="text" size="8"/></td>
          <td><input type="submit" jwcid="@Submit" value="Get Quote"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

The ***inputSymbol*** component is defined as the ***ValidField*** to edit the page's symbol property

Page Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.Home">
  <property-specification name="error" type="java.lang.String"/>
  <property-specification name="symbol" type="java.lang.String"/>

  <bean name="delegate" class="org.apache.tapestry.valid.ValidationDelegate"/>

  <bean name="validator" class="org.apache.tapestry.valid.StringValidator"
   lifecycle="page">
     <set-property name="required" expression="true"/>
     <set-property name="clientScriptingEnabled" expression="true"/>
   </bean>

  <component id="inputSymbol" type="ValidField">
    <binding name="value" expression="symbol"/>
    <binding name="validator" expression="beans.validator"/>
    <static-binding name="displayName" value="Enter Symbol:"/>
  </component>
 </page-specification>
```

Page Specification (cont.)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
```

Creates a bean named ***delegate*** that instantiates an instance of the ***ValidationDelegate*** class

This object is set as the ***delegate*** on our Form component

```
<
  <property-specification name="symbol" type="java.lang.String"/>

  <bean name="delegate" class="org.apache.tapestry.valid.ValidationDelegate"/>

  <bean name="validator" class="org.apache.tapestry.valid.StringValidator"
        lifecycle="page">
     <set-property name="required" expression="true"/>
     <set-property name="clientScriptingEnabled" expression="true"/>
   </bean>

  <component id="inputSymbol" type="ValidField">
    <binding name="value" expression="symbol"/>
    <binding name="validator" expression="beans.validator"/>
    <static-binding name="displayName" value="Enter Symbol:"/>
  </component>
 </page-specification>
```

Page Specification (cont.)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<pa
  <
  <
```

Creates a bean named ***validator*** that instantiates an instance of the ***StringValidator*** class

This object is used by the inputSymbol component to validate the user’s input

```
  <bean name="delegate" class="org.apache.tapestry.valid.ValidationDelegate"/>

  <bean name="validator" class="org.apache.tapestry.valid.StringValidator"
        lifecycle="page">
     <set-property name="required" expression="true"/>
     <set-property name="clientScriptingEnabled" expression="true"/>
   </bean>

  <component id="inputSymbol" type="ValidField">
    <binding name="value" expression="symbol"/>
    <binding name="validator" expression="beans.validator"/>
    <static-binding name="displayName" value="Enter Symbol:"/>
  </component>
 </page-specification>
```

Page Specification (cont.)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.quote.tapestry.Home">
  <property-specification name="error" type="java.lang.String"/>
  <property-specification name="symbol" type="java.lang.String"/>

  <bean name="delegate" class="org.apache.tapestry.valid.ValidationDelegate"/>

  <bean name="validator" class="org.apache.tapestry.valid.StringValidator"
```

The ***inputSymbol*** component is defined as the ***ValidField*** to edit the page's ***symbol*** property

```
     <set-property name="clientScriptingEnabled" expression="true"/>
   </bean>

  <component id="inputSymbol" type="ValidField">
    <binding name="value" expression="symbol"/>
    <binding name="validator" expression="beans.validator"/>
    <static-binding name="displayName" value="Enter Symbol:"/>
  </component>
 </page-specification>
```

Page Class

```
public class Home extends BasePage {
    …
    public IValidationDelegate getValidationDelegate() {
        return (IValidationDelegate) getBeans().getBean("delegate");
    }

    public void formSubmit(IRequestCycle cycle) {
        if (getValidationDelegate().getHasErrors()) {
            return;
        }
        …
    }
}
```

Page Class (cont.)

```
public class Home extends BasePage {
    …
    public IValidationDelegate getValidationDelegate() {
        return (IValidationDelegate) getBeans().getBean("delegate");
    }
```

Convenience method to lookup the delegate bean defined in the Page specification

```
          if (getValidationDelegate().getHasErrors()) {
              return;
          }
          …
     }
}
```

Page Class (cont.)

```
public class Home extends BasePage {
    …
    public IValidationDelegate getValidationDelegate() {
        return (IValidationDelegate) getBeans().getBean("delegate");
    }

    public void formSubmit(IRequestCycle cycle) {
        if (getValidationDelegate().getHasErrors()) {
            return;
        }
```

If the IValidationDelegate has reported a validation error then reactivate the current page so the user can see the feedback

```
}
```

i18n

- i18n in Tapestry is relatively simple
- Capability to i18n both text and images
- i18n page content by either adding a properties file for each page that contains localized messages or provide a i18n version of the template
- The approach of using a message properties file for each page is easier to manage and maintain than a huge application wide message properties file
- It may be useful to create i18n versions of the entire page template if pages differ across locales in more than just language, for example having different layouts or composed of different components

Properties file

```
title = Stock Quote Application
fetchError = Error While Retrieving Quote for {0}. \
             Please check symbol and try again.
enterSymbol = Enter Symbol:
```

HTML Template

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/style.css" title="style"/>
    <title>
      <span jwcid="@Insert" value="message:title">Stock Quote Application</span>
    </title>
  </head>
  <body jwcid="@Body">
    <h1>
      <span jwcid="@Insert" value="message:title">Stock Quote Application</span>
    </h1>
    …
  </body>
</html>
```

- The OGNL message: prefix instructs Tapestry to retrieve the value for the specified key from the Page's properties file

Page Specification

```
…
<page-specification class="com.ociweb.quote.tapestry.Home">
  …
  <component id="inputSymbol" type="ValidField">
    <binding name="value" expression="symbol"/>
    <binding name="validator" expression="beans.validator"/>
    <message-binding name="displayName" key="enterSymbol"/>
  </component>

 </page-specification>
```

- The message-binding element looks up the value specified by the key attribute

Page Class

```
public abstract class Home extends BasePage {
    …

    public void formSubmit(IRequestCycle cycle) {
       …
        try {
            …
        } catch (QuoteFetchException e) {
            String error = format("fetchError", symbol);
            setError(error);
        }
    }
}
```

- The format method formats the message with the specified key with the specified parameter(s) (works similar to `java.text.MessageFormat`)

Tapestry Components

- Tapestry ships with more than 40 components ready for you to use when building your apps
- The contrib package that ships with Tapestry includes an additional 20 components including the contrib:table component
- To see some of the Tapestry components at work visit the Tapestry Component Workbench at: http://www.t-deli.com/app
- Creating your own Tapestry components is similar to creating pages in a Tapestry application

Built-in Tapestry Components

- Control oriented
  - Foreach, Conditional
- Form
  - Form, TextField, Checkbox, Select, Radio, Upload, ValidField, DatePicker
- Link based
  - DirectLink, PageLink, ExternalLink

Tapestry Workbench Demo

- View a demo of the Tapestry workbench demo

Creating Tapestry Components

- A typical Tapestry component consists of a:
  - A component specification (XML document with .jwc extension)
  - HTML component template (optional)
  - A Java class that implements the `org.apache.tapestry.IComponent` interface (optional)
- Creating custom components allows you to adhere to the DRY principal

Create ValidationError component

- The ValidationError component will be used to to display a validation error message for invalid form input
- This component will be built using “built-in” Tapestry components

ValidationError Template

```
<div jwcid="@Conditional" condition="ognl:delegate.hasErrors">
  <div class="error">
    <table>
      <tr><td>
          <span jwcid="@Delegator"
                delegate="ognl:delegate.firstError">
              Error Message
          </span>
        </td></tr>
    </table>
  </div>
</div>
```

ValidationError Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE component-specification PUBLIC
  "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
  "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<component-specification class="org.apache.tapestry.BaseComponent"
                         allow-informal-parameters="no">
  <parameter name="delegate"
             type="org.apache.tapestry.valid.IValidationDelegate"
             required="yes" direction="in"/>
</component-specification>
```

ValidationError Usage

Before:

```
<html>
  …
  <div jwcid="@Conditional" condition="ognl:beans.delegate.hasErrors">
    <div class="error">
      <table><tr><td>
        <span class="error">
          <span class="error" jwcid="@Delegator”
                delegate="ognl:beans.delegate.firstError">
            Validation Error Message
          </span></span></td></tr></table>
    </div></div>
  …
</html>
```

After:

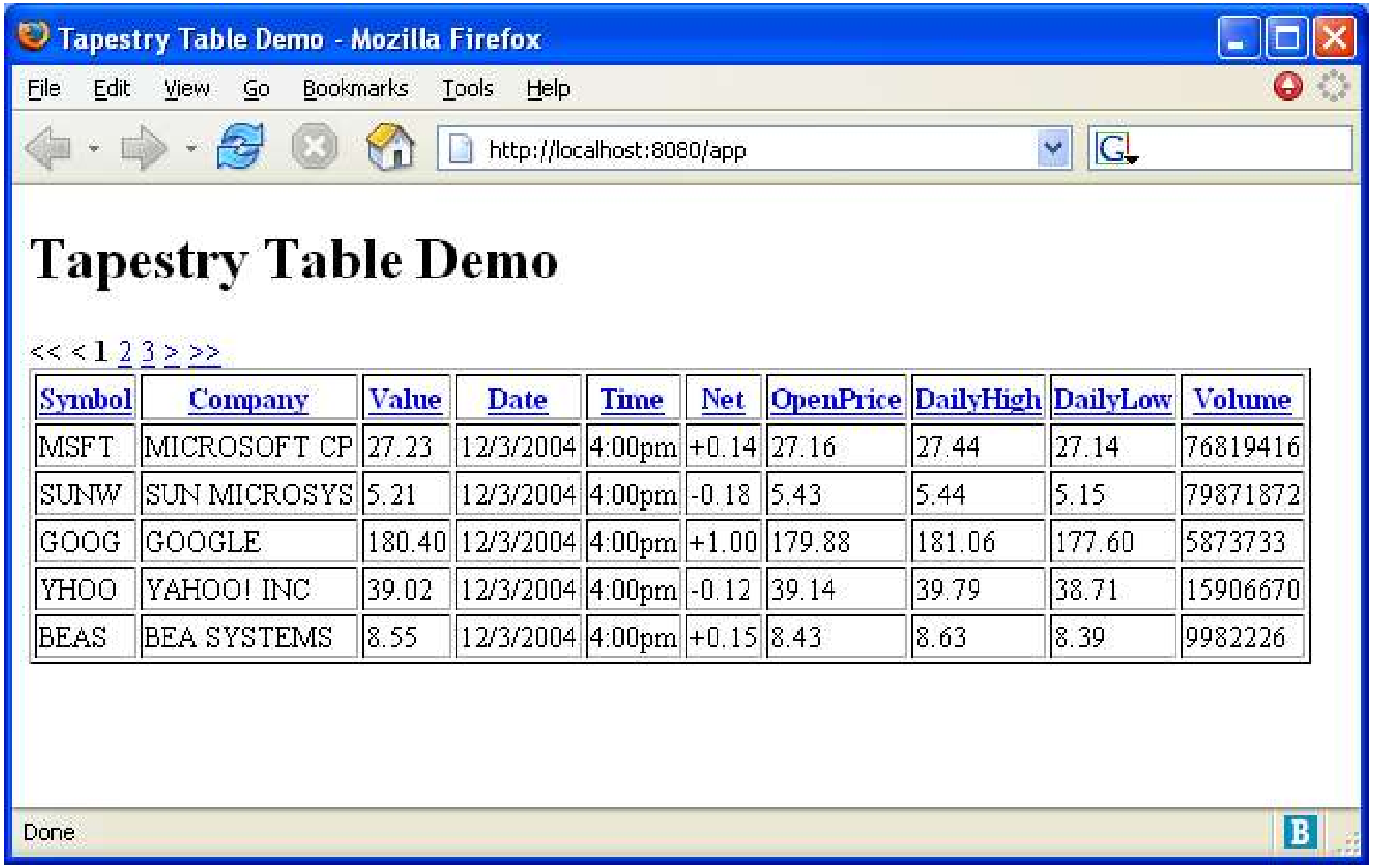
```
<html>
  …
    <span jwcid="@ValidationError" delegate="ognl:beans.delegate"/>
  …
</html>
```

This component will help us adhere to the DRY principal in an application with multiple input forms

contrib:table Example

- contrib:table renders a HTML table element
- Presents a sortable and pageable table

contrib:table Screenshot


Tapestry Table Demo - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/app
Done

Tapestry Table Demo

<< < 1 2 3 > >>

Symbol	Company	Value	Date	Time	Net	OpenPrice	DailyHigh	DailyLow	Volume
MSFT	MICROSOFT CP	27.23	12/3/2004	4:00pm	+0.14	27.16	27.44	27.14	76819416
SUNW	SUN MICROSYS	5.21	12/3/2004	4:00pm	-0.18	5.43	5.44	5.15	79871872
GOOG	GOOGLE	180.40	12/3/2004	4:00pm	+1.00	179.88	181.06	177.60	5873733
YHOO	YAHOO! INC	39.02	12/3/2004	4:00pm	-0.12	39.14	39.79	38.71	15906670
BEAS	BEA SYSTEMS	8.55	12/3/2004	4:00pm	+0.15	8.43	8.63	8.39	9982226

HTML Template

```
<html …>
<body …>
  …
  <table class="tapestry-table"
      jwcid="table@contrib:Table"
      source="ognl:quotes"
      columns="Symbol, Company, Value,
               Date, Time, Net, OpenPrice,
               DailyHigh, DailyLow, Volume"
      pageSize="5">
  </table>
</body>
</html>
```

The ***source*** parameter specifies the source of the data for the table.

The ***columns*** parameter specifies the column names for the table

The ***pageSize*** parameter specifies the number of records per page

Page Specification

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page-specification
      PUBLIC "-//Apache Software Foundation//Tapestry Specification 3.0//EN"
      "http://jakarta.apache.org/tapestry/dtd/Tapestry_3_0.dtd">
<page-specification class="com.ociweb.tapestry.table.TableDemo">
    <property-specification name="quotes" type="java.util.List"
            persistent="yes"/>
</page-specification>
```

The property must be ***persistent*** to allow the data to be sorted and paged

Page Class

```
public abstract class TableDemo extends BasePage implements PageRenderListener {
    …
    public abstract List getQuotes();
    public abstract void setQuotes(List dataItems);

    public void pageBeginRender(PageEvent event) {
        List list = getQuotes();
        if (list == null) {
            setQuotes(buildQuotes());
        }
    }

    private List buildQuotes() {
        List quotes = new ArrayList();
        …
        return quotes;
    }
}
```

The ***pageBeginRender*** method is defined by the ***PageRenderListener*** interface and is called when the page begins to render. It allows us a chance to initialize the quotes property.

Tools

Spindle

- Spindle (http://spindle.sourceforge.net/) is an Eclipse plug-in for Tapestry development
- Spindle 3.1 is a native Eclipse 3.0 feature
- Streamlines Tapestry development through the use of custom wizards and editors
- Helps developer by identifying errors at build time that normally would not be caught until runtime, such as invalid OGNL expressions and invalid component IDs.

IntelliJ IDEA

- Can create custom file templates for Page Specifications, Page classes, etc.

Summary

Strengths

- Developing web applications with Tapestry is simple yet elegant
- Allows you to write web applications without being concerned with the operation centric Servlet API
- Tapestry page templates are standard HTML with a few special attributes and tags recognized by Tapestry
- HTML WYSIWYG editors can be used to edit and preview page templates
- Tapestry does not require a monolithic, application-wide configuration file
- Event handler driven
- Line-precise error reporting

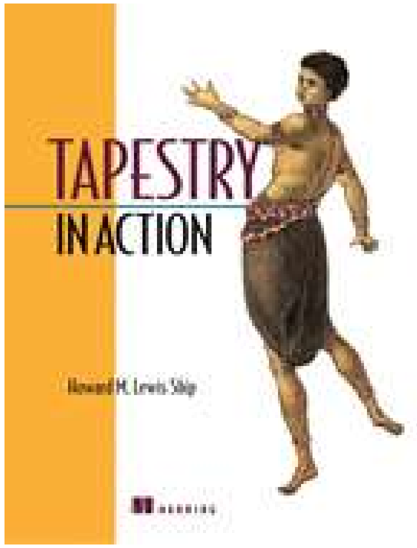
Summary (cont.)

Weaknesses

- Not widely accepted in the Java community
- Not a "standard"
  - Not J2EE standard like JSF
  - Not the "de-facto" standard like Struts
  - Results should be more important than standards
- Ugly URLs
  - Makes J2EE declarative security impossible
  - There's a Friendly URLs patch on the Tapestry wiki to fix this
- Difficult to Unit Test Page classes because they are abstract
  - Tapestry Test Assist can help solve this problem

Resources

- Home Page - http://jakarta.apache.org/tapestry/
- Tapestry In Action - http://www.manning.com/lewisship/
  - ISBN - 1932394117



Resources (cont.)

- Introduction to Jakarta Tapestry - http://www.ociweb.com/jnb/jnbMay2004.html
- Tapestry Wiki -http:// jakarta.apache.org/tapestry/wiki_frame.html
- OGNL page - http://www.ognl.org/
- Tapestry Component Reference - http://jakarta.apache.org/tapestry/doc/ComponentReference/ index.html
- Tapestry Component Workbench – http://www.tdeli.com/app

Jakarta Tapestry – Q & A

Rob Smith
Senior Software Engineer
Object Computing, Inc.
St. Louis, MO