



JSF: Data Tables

Originals of Slides and Source Code for Examples:
<http://www.coreservlets.com/JSF-Tutorial/>

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live training on JSF 1.x or 2.0, please see courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
 - Java 5, Java 6, intermediate/beginning servlets/JSP, advanced servlets/JSP, Struts, JSF, Ajax, GWT, custom courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- **Motivation**
- **Basic syntax**
- **Defining table headings**
- **Formatting tables with style sheets**
- **Displaying database tables**

4

Motivation: Dealing with Variable Length Data

- **Issue**
 - What if the business/data-access logic creates something with an indeterminate number of elements?
 - How do you output it in the final JSP page without breaking the MVC model?
- **Alternatives (without data tables)**
 - Non-looping
 - `<h:outputText value="#{bankCustomer.depositTable}" escape="true"/>`
 - `<mytags:showDeposits custID="..." styleClasses="..."/>`
 - Looping
 - JSTL
 - `<% for(...) { ... %>`
HTML Code
`<% } %>`

5

JSP Constructs for Handling Variable-Length Data

Simplest
for Page
Author



Most
Control
for Page
Author

- Bean
- Custom tag
- **h:dataTable**
- t:dataList
- JSTL loop
- JSP scripting loop

6

© 2009 Marty Hall



Basic Syntax: h:dataTable and h:column

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Building HTML Tables

- **Assume you need HTML exposed in JSP page, so you cannot use the non-looping alternatives**
 - Using JSP scripting elements to loop is unwieldy
 - Using JSTL to loop is an option, but is still complex
- **JSF provides h:dataTable**
 - You give *one* row definition, and JSF repeats it for you

```
<h:dataTable value="#{someBean.someCollection}"
             var="rowVar"
             border="1">
  <h:column>
    <h:outputText value="#{rowVar.col1Data}"/>
  </h:column>
  <h:column>
    <h:outputText value="#{rowVar.col2Data}"/>
  </h:column>
  ...
</h:dataTable>
```

8

Details: h:dataTable

- **value: a collection of data (list of beans, usually). Legal collection types:**
 - Array
 - List (e.g., ArrayList, LinkedList)
 - ResultSet (must be scroll-insensitive: see later section)
 - Result (wrapped ResultSet from JSTL)
 - DataModel (in javax.faces.model)
- **var: bound to each collection entry in turn**
 - This entry should be something the JSF EL can output
 - Bean, array, List, Map
- **Other attributes**
 - Standard TABLE attributes
 - border, bgcolor, width, cellpadding, cellspacing, frame, ...
 - Style sheet designators
 - rowClasses, headerClass, footerClass

9

Details: h:column

- **Usually encloses h:outputText elements**
 - Which reference the variable from the enclosing h:dataTable
- ```
<h:column>
 <h:outputText value="#{rowVar.colData}"/>
</h:column>
```
- **Can enclose other h:Xxxx elements**
    - h:outputText, h:inputText, etc.
  - **Regular HTML must be enclosed in f:verbatim**
- ```
<h:column>  
  <f:verbatim>First Name: </f:verbatim>  
  <h:outputText value="#{rowVar.firstName}"/>  
</h:column>
```
- **Table headings and footers specified with f:facet**
 - See later example

10

Example: Printing Table of Sales Based on Array

- **SalesBean class represents sales of apples and oranges (in dollars)**
- **Array of SalesBean objects represents quarterly sales in year**
- **All values to be presented in HTML table**

11

SalesBean: Basic Code

```
public class SalesBean {
    private double apples = 0.0, oranges = 0.0;

    public SalesBean() {}

    public SalesBean(double apples, double oranges) {
        setApples(apples);
        setOranges(oranges);
    }

    public double getApples() { return(apples); }

    public void setApples(double apples) {
        this.apples = apples;
    }

    public double getOranges() { return(oranges); }

    public void setOranges(double oranges) {
        this.oranges = oranges;
    }
}
```

12

SalesBean: Code to Create Sample Sales Array

```
public SalesBean[] getYearlySales() {
    SalesBean[] yearlySales =
        { new SalesBean(100.22, 200.32),
          new SalesBean(300.44, 400.55),
          new SalesBean(500.66, 600.77),
          new SalesBean(700.88, 800.99) };
    return(yearlySales);
}
```

13

faces-config.xml

```
...
<faces-config>
  <managed-bean>
    <managed-bean-name>salesBean</managed-bean-name>
    <managed-bean-class>
      coreservlets.SalesBean
    </managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
  </managed-bean>
  ...
</faces-config>
```

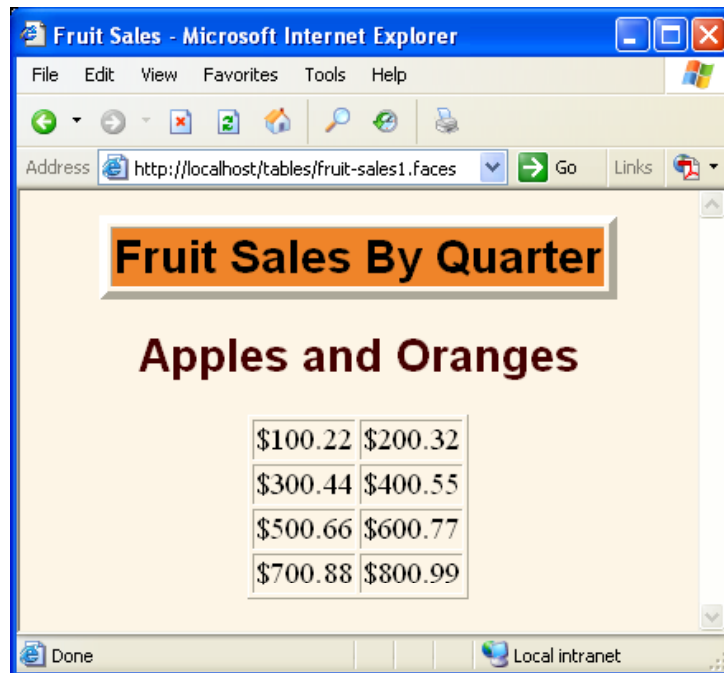
14

fruit-sales1.jsp

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<f:view>
  ...
  <H2>Apples and Oranges</H2>
  <h:dataTable value="#{salesBean.yearlySales}"
    var="quarterlySales"
    border="1">
    <h:column>
      <f:verbatim>${}</f:verbatim>
      <h:outputText value="#{quarterlySales.apples}"/>
    </h:column>
    <h:column>
      <f:verbatim>${}</f:verbatim>
      <h:outputText value="#{quarterlySales.oranges}"/>
    </h:column>
  </h:dataTable>
  ...
</f:view>
```

15

fruit-sales1.faces



The screenshot shows a Microsoft Internet Explorer browser window displaying a web page. The address bar shows the URL `http://localhost/tables/fruit-sales1.faces`. The page content includes a title "Fruit Sales By Quarter" in a blue box, followed by the subtitle "Apples and Oranges". Below this is a table with four rows of sales data.

\$100.22	\$200.32
\$300.44	\$400.55
\$500.66	\$600.77
\$700.88	\$800.99

16

© 2009 Marty Hall



Defining Table Headings

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Headers and Footers

- **Headers**

- Use f:facet with name="header"
- Value can be f:verbatim or h:outputText
- Still need h:outputText for non-heading value

- **Footers**

- Use f:facet with name="footer"
- Value can be f:verbatim or h:outputText

- **Example Code**

```
<h:column>
  <f:facet name="header">
    <f:verbatim>...</f:verbatim>
  </f:facet>
  <h:outputText value="#{rowVar.colVal}"/>
</h:column>
```

18

Example: Displaying List of Monthly Sales

- **MonthlySalesBean class**

- Represents a month name and a SalesBean for the apples/oranges sales in that month

- **ArrayList represents sales for January through December**

- **Want headings to be "Month", "Apples", and "Oranges"**

- **Also want to use textfields to display sales numbers, so that they can be edited**

- Point is to illustrate that tables can contain input elements
- Backend code to process results omitted in this example

19

MonthlySalesBean: Basic Code

```
public class MonthlySalesBean {
    private String month;
    private SalesBean salesBean;

    public MonthlySalesBean() {}

    public MonthlySalesBean(String month,
                             double apples,
                             double oranges) {
        setMonth(month);
        setSalesBean(new SalesBean(apples, oranges));
    }

    public String getMonth() { return(month); }

    public void setMonth(String month) {
        this.month = month; }

    public SalesBean getSalesBean() {
        return(salesBean);
    }
    public void setSalesBean(SalesBean salesBean) {
        this.salesBean = salesBean; }
}
```

20

MonthlySalesBean: Code to Give List of Sales Figures

```
public List<MonthlySalesBean> getYearlySales() {
    List<MonthlySalesBean> yearlySales =
        new ArrayList<MonthlySalesBean>();
    yearlySales.add
        (new MonthlySalesBean("January", 101.11, 501.11));
    yearlySales.add
        (new MonthlySalesBean("February", 102.22, 502.22));
    yearlySales.add
        (new MonthlySalesBean("March", 103.33, 503.33));
    ...
    yearlySales.add
        (new MonthlySalesBean("November", 111.11, 511.11));
    yearlySales.add
        (new MonthlySalesBean("December", 112.22, 512.22));
    return(yearlySales);
}
```

21

faces-config.xml

```
...
<faces-config>
  <managed-bean>
    <managed-bean-name>
      monthlySalesBean
    </managed-bean-name>
    <managed-bean-class>
      coreservlets.MonthlySalesBean
    </managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
  </managed-bean>
  ...
</faces-config>
```


22

fruit-sales2.jsp

```
<h:form>
<h:dataTable value="#{monthlySalesBean.yearlySales}"
  var="monthlySales"
  border="1">
  <h:column>
    <f:facet name="header">
      <f:verbatim>Month</f:verbatim>
    </f:facet>
    <h:outputText value="#{monthlySales.month}"/>
  </h:column>
  <h:column>
    <f:facet name="header">
      <f:verbatim>Apples</f:verbatim>
    </f:facet>
    <f:verbatim>${}</f:verbatim>
    <h:inputText
      value="#{monthlySales.salesBean.apples}"/>
  </h:column>
  ... Oranges ...
</h:dataTable>
</h:form>
```

23

fruit-sales2.faces



The screenshot shows a Microsoft Internet Explorer window titled "Fruit Sales - Microsoft Internet Explorer". The address bar displays "http://localhost/tables/fruit-sales2.faces". The main content area features a table titled "Fruit Sales By Month" with three columns: "Month", "Apples", and "Oranges". The table lists sales data for each month from January to December. Below the table is an "Update Sales" button. The browser's status bar at the bottom shows "Done" and "Local intranet".

Month	Apples	Oranges
January	\$101.11	\$501.11
February	\$102.22	\$502.22
March	\$103.33	\$503.33
April	\$104.44	\$504.44
May	\$105.55	\$505.55
June	\$106.66	\$506.66
July	\$107.77	\$507.77
August	\$108.88	\$508.88
September	\$109.99	\$509.99
October	\$110.11	\$510.11
November	\$111.11	\$511.11
December	\$112.22	\$512.22

24

© 2009 Marty Hall



Formatting Tables with Style Sheets

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Options

- **Use explicit formatting in JSP**
 - Requires `f:verbatim` before and after each entry
 - Long and clumsy
- **Embed formatting in bean**
 - Not accessible to Web developer
 - Risks inconsistencies with style sheet
 - Beans are for model, not view
- **Use `rowClasses`, `headerClass`, `footerClass`**
 - **`rowClasses`**: comma-separated list of CSS styles. Applied to each row until list ends, then repeats
 - **`headerClass`**: CSS style for heading
 - **`footerClass`**: CSS style for footer

26

Example: Formatted Table of Monthly Sales

- **Same `MonthlySalesBean`**
- **Same `faces-config.xml` entry**
- **Style applied to heading**
- **Two styles applied to rows**
 - Results in alternating styles

27

Style Sheet (styles.css)

```
.HEADING {  
  font-family: Arial, Helvetica, sans-serif;  
  font-weight: bold;  
  font-size: 20px;  
  color: black;  
  background-color: silver;  
  text-align: center;  
}
```

28

Style Sheet (styles.css)

```
.ROW1 {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 18px;  
  color: black;  
  background-color: white;  
  text-indent: 20px;  
}  
.ROW2 {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 18px;  
  color: white;  
  background-color: black;  
  text-indent: 20px;  
}
```


29

fruit-sales.jsp

```
<h:dataTable value="#{monthlySalesBean.yearlySales}"
  var="monthlySales"
  border="1"
  headerClass="HEADING"
  rowClasses="ROW1,ROW2">
  <h:column>
    <f:facet name="header">
      <f:verbatim>Month</f:verbatim>
    </f:facet>
    <h:outputText value="#{monthlySales.month}"/>
  </h:column>
  <h:column>
    ... Apples ...
  </h:column>
  <h:column>
    ... Oranges ...
  </h:column>
</h:dataTable>
```

30

fruit-sales3.faces



The screenshot shows a web browser window titled "Fruit Sales - Microsoft Internet Explorer". The address bar displays "http://localhost/tables/fruit-sales3.faces". The main content area features a table with the following data:

Month	Apples	Oranges
January	\$101.11	\$501.11
February	\$102.22	\$502.22
March	\$103.33	\$503.33
April	\$104.44	\$504.44
May	\$105.55	\$505.55
June	\$106.66	\$506.66
July	\$107.77	\$507.77
August	\$108.88	\$508.88
September	\$109.99	\$509.99
October	\$110.11	\$510.11
November	\$111.11	\$511.11
December	\$112.22	\$512.22

31



Database-Driven Tables

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Options for Displaying Database Results

- **Extract data from result set and place in array or List**
 - Tedious
 - Requires a bean to represent a row of data
- **Use ResultSet as arg to value of h:dataTable**
 - Should work, according to spec
 - In practice, you must use scroll-insensitive or it crashes
 - ResultSet is connected, so a big pain to go back and close connections (or return them to pool) later
- **Use JSTL ResultSupport class to turn ResultSet into Result**
 - Only one extra line of code
 - Robust and reliable
 - Result is disconnected, so you can close connection or return it to pool before using the Result

Example: Display Customers from Northwind Database

- Store connection info in server.xml for Tomcat
- Retrieve Connection from JNDI
- Wrap ResultSet in a Result
- Pass Result to h:dataTable
- Use column names instead of bean property names

34

NorthwindTableBean: Basic Code

```
package coreservlets;
import java.sql.*;
import javax.sql.*;
import javax.naming.*;
import javax.servlet.jsp.jstl.sql.*;

public class NorthwindTableBean {
    private String tableName = "customers";

    public String getTableName() {
        return(tableName);
    }

    public void setTableName(String tableName) {
        this.tableName = tableName;
    }
}
```

35

NorthwindTableBean: Code to Perform Query and Return Result

```
public Result getTable() {
    try {
        Context context = new InitialContext();
        DataSource dataSource =
            (DataSource)context.lookup
                ("java:comp/env/jdbc/Northwind");
        Connection connection =
            dataSource.getConnection();
        Statement statement =
            connection.createStatement();
        String query = "SELECT * from " + getTableName();
        ResultSet resultSet =
            statement.executeQuery(query);
        return(ResultSupport.toResult(resultSet));
    } catch(Exception e) { return(null); }
}
```

36

faces-config.xml

```
...
<faces-config>
    <managed-bean>
        <managed-bean-name>
            northwindBean
        </managed-bean-name>
        <managed-bean-class>
            coreservlets.NorthwindTableBean
        </managed-bean-class>
        <managed-bean-scope>request</managed-bean-scope>
    </managed-bean>...
</faces-config>
```

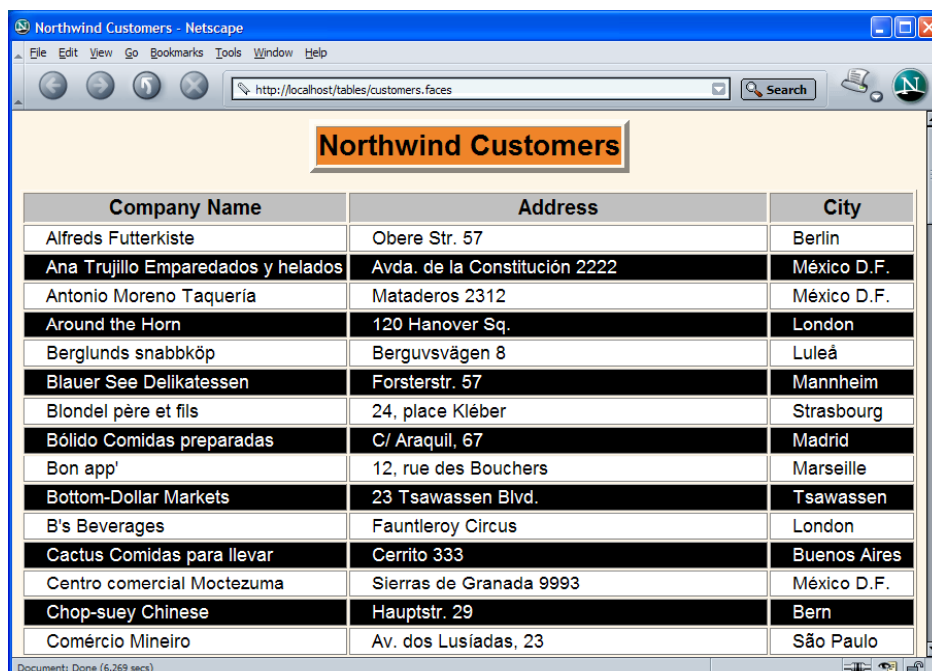
37

customers.jsp

```
<h:dataTable value="#{northwindBean.table}"
             var="dbRow"
             border="1"
             headerClass="HEADING"
             rowClasses="ROW1,ROW2">
  <h:column>
    <f:facet name="header">
      <f:verbatim>Company Name</f:verbatim>
    </f:facet>
    <h:outputText value="#{dbRow.CompanyName}"/>
  </h:column>
  <h:column>
    <f:facet name="header">
      <f:verbatim>Address</f:verbatim>
    </f:facet>
    <h:outputText
      value="#{dbRow.Address}"/>
  </h:column>
  ... City ...
</h:dataTable>
```

38

customers.faces



Northwind Customers

Company Name	Address	City
Alfreds Futterkiste	Obere Str. 57	Berlin
Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
Antonio Moreno Taquería	Mataderos 2312	México D.F.
Around the Horn	120 Hanover Sq.	London
Berglunds snabbköp	Berguvsvägen 8	Luleå
Blauer See Delikatessen	Forsterstr. 57	Mannheim
Blondel père et fils	24, place Kléber	Strasbourg
Bólido Comidas preparadas	C/ Araquil, 67	Madrid
Bon app'	12, rue des Bouchers	Marseille
Bottom-Dollar Markets	23 Tsawassen Blvd.	Tsawassen
B's Beverages	Fauntleroy Circus	London
Cactus Comidas para llevar	Cerrito 333	Buenos Aires
Centro comercial Moctezuma	Sierras de Granada 9993	México D.F.
Chop-suey Chinese	Hauptstr. 29	Bern
Comércio Mineiro	Av. dos Lusíadas, 23	São Paulo

Document: Done (6.269 secs)

39

Summary

- **<h:dataTable value="collection" var="variable" ...>**
 - Collection usually array, List, or Result
 - HTML-oriented attributes:
 - border, bgcolor, width, etc.
 - Style sheet designators
 - rowClasses, headerClass, footerClass
- **<h:column>...</h:column>**
 - Encloses code that refers to variable
 - Usually h:outputText, but can be any JSF component
 - f:verbatim for literal text
 - f:facet for headings and footer

40

© 2009 Marty Hall



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.