© 2012 Marty Hall



Advanced Swing & MVC Custom Data Models and Cell Renderers

Originals of Slides and Source Code for Examples: http://courses.coreservlets.com/Course-Materials/java.html

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



Agenda

- Building a simple static JList
- Adding and removing entries from a JList at runtime

Making a custom data model

 Telling JList how to extract data from existing objects

Making a custom cell renderer

 Telling JList what GUI component to use for each of the data cells

MVC Architecture

Custom data models

 Changing the way the GUI control obtains the data. Instead of copying data from an existing object into a GUI control, simply tell the GUI control how to get at the existing data.

Custom cell renderers

 Changing the way the GUI control displays data values. Instead of changing the data values, simply tell the GUI control how to build a Swing component that represents each data value.

Main applicable components

- JList
- JTable
- JTree



Simple JList: Example Code



Simple JList: Example Output

Enti	rv 1 _
Ent	v 2 ≡
Ent	
Ent	
LIU	y 4 -
JList Selection	
T	E t 2



Changeable JList: Example Code

```
String[] entries = { "Entry 1", "Entry 2", "Entry 3",
                      "Entry 4", "Entry 5", "Entry 6" };
sampleModel = new DefaultListModel();
for(int i=0; i<entries.length; i++) {
    sampleModel.addElement(entries[i]);
}
sampleJList = new JList(sampleModel);
sampleJList.setVisibleRowCount(4);
Font displayFont = new Font("Serif", Font.BOLD, 18);
sampleJList.setFont(displayFont);
JScrollPane listPane = new JScrollPane(sampleJList);
```



Changeable JList: Example Output

	Entry 7 [^]	
	Entry 8	
	Entry 9	
	Entry 10 -	
Addina Entr	es	
free states and states		
Add	Entry to Bottom of JI	List
	13	



Custom Model: Example Code

public class JavaLocationListModel implements ListModel {
 private JavaLocationCollection collection;



- Country, comment, flag file

JList with Custom Model: Example Code

JavaLocationCollection collection =
 new JavaLocationCollection();
JavaLocationListModel listModel =
 new JavaLocationListModel(collection);
JList sampleJList = new JList(listModel);
Font displayFont =
 new Font("Serif", Font.BOLD, 18);
sampleJList.setFont(displayFont);
content.add(sampleJList);

JList with Custom Model: Example Output

JList with a Custom Data Model
Java, Belgium (near Liege).
Java, Brazil (near Salvador).
Java, Colombia (near Bogota).
Java, Indonesia (main island).
Java, Jamaica (near Spanish Town).
Java, Mozambique (near Sofala).
Java, Philippines (near Quezon City).
Java, Sao Tome (near Santa Cruz).
Java, Spain (near Viana de Bolo).
Java, Suriname (near Paramibo).
Java, United States (near Montgomery, Alabama).
Java, United States (near Dallas, Texas).

JList with Custom Cell Renderer

Idea

 Instead of predetermining how the JList will draw the list elements, Swing lets you specify what graphical component to use for the various entries.
 Attach a ListCellRenderer that has a getListCellRendererComponent method that determines the GUI component used for each cell.

getListCellRendererComponent arguments

- JList: the list itself
- Object: the value of the current cell
- int: the index of the current cell
- boolean: is the current cell selected?
- boolean: does the current cell have focus?

Custom Renderer: Example Code

```
public class JavaLocationRenderer extends
                                  DefaultListCellRenderer {
  private Map<Object,ImageIcon> iconTable =
    new HashMap<Object,ImageIcon>();
  public Component getListCellRendererComponent
                (JList list, Object value, int index,
                 boolean isSelected, boolean hasFocus) {
    JLabel label = (JLabel)super.getListCellRendererComponent
                       (list,value,index,isSelected,hasFocus);
    if (value instanceof JavaLocation) {
      JavaLocation location = (JavaLocation)value;
      ImageIcon icon = iconTable.get(value);
      if (icon == null) {
        icon = new ImageIcon(location.getFlagFile());
        iconTable.put(value, icon);
      }
      label.setIcon(icon);
    return(label);
```

Custom Renderer: Example Output



Summary

Simple static JList

- Pass array of strings to JList constructor

Simple changeable JList

- Pass DefaultListModel to JList constructor.
- Add/remove data to/from the model, not the JList.

Custom data model

- Have real data implement ListModel interface.
- Pass real data to JList constructor.

Custom cell renderer

- Assign a ListCellRenderer
- ListCellRenderer has a method that determines the
 - Component to be used for each cell



© 2012 Marty Hall

Questions?

JSF 2, PrimeFaces, Java 7, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training.

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.