



DataDirect® SQLServer JDBC
Built-in bulk load and Windows Authentication from Unix/Linux

Progress
Try now

[ANDROID ▾](#)
[CORE JAVA ▾](#)
[DESKTOP JAVA ▾](#)
[ENTERPRISE JAVA ▾](#)
[JAVA BASICS ▾](#)
[JVM LANGUAGES ▾](#)
[SOFTWARE DEVELOPMENT ▾](#)
[DEVOPS ▾](#)

Home » Enterprise Java » jsf » Multiple Selection ListBox Example with JSF 2.0

ABOUT THODORIS BAIS

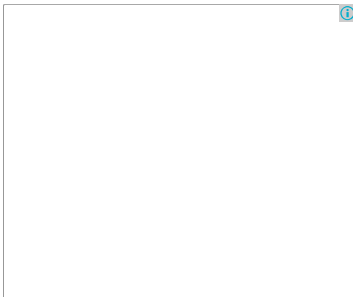


Thodoris is an Oracle Certified Associate Java Programmer and currently works as a Junior Software Developer, for Intrasoftware International S.A. He holds a diploma at Informatics & Telecommunications Engineering and is interested in continuous development.



Multiple Selection ListBox Example with JSF 2.0

Posted by: Thodoris Bais in jsf February 24th, 2014



Hello again World, it's been a long time since my last example, so let's get back to work! If you can't remember, in our last example, we were talking about listboxes integration with JSF 2.0. Ok, but what happens if user has to select more than one single option from a listbox? That is, that's the purpose of this example.

Want to be a JSF Ninja?

Subscribe to our newsletter and download the JSF 2.0 Programming Cookbook right now!

In order to get you prepared for your JSF development needs, we have compiled numerous recipes to help you kick-start your projects. Besides reading them online you may download the eBook in PDF format!

Email address:

Sign up

In JSF, we can use the

```
<h:selectManyListbox />
```

tag, in order to render a multiple-selection listbox. The way that we can display a multi-selectable HTML listbox that includes three options, could be displayed as we saw in the previous example; the related HTML rendering is very accurate:

```
1 <select name="j_idt6:j_idt8" multiple="multiple" size="3">
2   <f:selectItem itemValue="2000" itemLabel="Year : 2000" />
3   <f:selectItem itemValue="2010" itemLabel="Year : 2010" />
4   <f:selectItem itemValue="2020" itemLabel="Year : 2020" />
5 </select>
```

The three ways used to render a multiple selection listbox are similar to the single selection listbox:

1. Hardcoded value in a

```
f:selectItem
```

tag.

2. Generated values using a Map and passed into the fore-mentioned tag.

3. Generate values using an Object Array and passed again into the

```
f:selectItem
```

tag, then represent the value using a

```
var
```

attribute.

1. Backing Bean

We have to hold the values of our listboxes in a

```
collection
```

or an

Google

Descubra o que o Google pode fazer pelo seu negócio

Juntos dá pra fazer mais

CLIQUE E DESCUBRA

Google

Descubra o que o Google pode fazer pelo seu negócio

Juntos dá pra fazer mais

CLIQUE E DESCUBRA



Find Top Developers Fast

The search is over. Let Toptal match you with proven developers.

NEWSLETTER

169,847 insiders are already enjoying weekly updates and complimentary whitepapers!

Join them now to gain exclusive access to the latest news in the Java world, as well as insights about Android, Scala, Groovy and other related technologies.

Email address:

Sign up

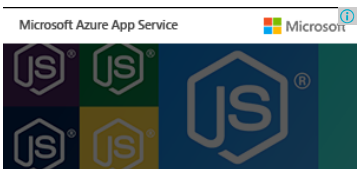
JOIN US



With **1,240,600** monthly unique visitors and over **500** authors we are placed among the top Java related sites around. Constantly being on the lookout for partners; we encourage you to join us. So If you have a blog with

unique and interesting content then you should check out our **JCG** partners program. You can also

be a **guest writer** for Java Code Geeks and hone your writing skills!



Array

, otherwise we'll get an error.

UserBean.java

```

001 package com.javacodegeeks.enterprise.jsf;
002
003 import java.io.Serializable;
004 import java.util.Arrays;
005 import java.util.LinkedHashMap;
006 import java.util.Map;
007
008 import javax.faces.bean.ManagedBean;
009 import javax.faces.bean.SessionScoped;
010
011 @ManagedBean
012 @SessionScoped
013 public class UserBean implements Serializable{
014
015     private static final long serialVersionUID = 4256272866128337548L;
016
017     public String[] favoriteCar1;
018     public String[] favoriteCar2;
019     public String[] favoriteCar3;
020
021     public String[] getFavoriteCar1() {
022         return favoriteCar1;
023     }
024     public void setFavoriteCar1(String[] favoriteCar1) {
025         this.favoriteCar1 = favoriteCar1;
026     }
027     public String[] getFavoriteCar2() {
028         return favoriteCar2;
029     }
030     public void setFavoriteCar2(String[] favoriteCar2) {
031         this.favoriteCar2 = favoriteCar2;
032     }
033     public String[] getFavoriteCar3() {
034         return favoriteCar3;
035     }
036     public void setFavoriteCar3(String[] favoriteCar3) {
037         this.favoriteCar3 = favoriteCar3;
038     }
039
040     public String getFavoriteCar1InString()
041     {
042         return Arrays.toString(favoriteCar1);
043     }
044     public String getFavoriteCar2InString()
045     {
046         return Arrays.toString(favoriteCar2);
047     }
048     public String getFavoriteCar3InString()
049     {
050         return Arrays.toString(favoriteCar3);
051     }
052
053     //generated by map
054     private static Map<String, Object> favoriteCar2Value;
055     static
056     {
057         favoriteCar2Value = new LinkedHashMap<String, Object>();
058
059         favoriteCar2Value.put("BMW Series 1 - 116", "116"); //label, value
060         favoriteCar2Value.put("BMW Series 1 - 118", "118");
061         favoriteCar2Value.put("BMW Series 1 - 120", "120");
062     }
063
064     public Map<String, Object> getFavoriteCar2Value()
065     {
066         return favoriteCar2Value;
067     }
068
069     //generated by object array
070     public static class Car
071     {
072         public String carLabel;
073         public String carValue;
074
075         public Car(String carLabel, String carValue)
076         {
077             this.carLabel = carLabel;
078             this.carValue = carValue;
079         }
080
081         public String getCarLabel()
082         {
083             return carLabel;
084         }
085
086         public String getCarValue()
087         {
088             return carValue;
089         }
090     }
091
092     public Car[] car3List;
093
094     public Car[] getFavoriteCar3Value()
095     {
096         car3List = new Car[3];
097
098         car3List[0] = new Car("BMW Series 3 - 316", "316");
099         car3List[1] = new Car("BMW Series 3 - 318", "318");
100         car3List[2] = new Car("BMW Series 3 - 320", "320");
101
102         return car3List;
103     }
104
105 }

```

2. Our JSF Pages

The welcome page:

index.xhtml



Aplicativos Get Node.js
criados com mais rapidez.

Confira aqui um material exclusivo
que você pode acessar gratuitamente.

CAREER OPPORTUNITIES

Jr. Software Developer **Techport Thirteen, Inc.**
Richmond, VA
Oct, 13

Jr. Software Developer **Techport Thirteen, Inc.**
Duluth, GA
Oct, 05

Java Developer **Transamerica**
Plano, TX
Oct, 05

Sr. Java Developer - Struts/Hibernate **Zenosys**
Trenton, NJ
Oct, 10

Java Software Engineer - Entry Level - Java
REQUIRED **LexisNexis**
King of Prussia, PA
Oct, 08

Java Software Engineer - Entry Level - Java
REQUIRED **Reed Elsevier**
King of Prussia, PA
Oct, 08

Web Developer - Entry Level **Verizon**
Alpharetta, GA
Sep, 26

Java Developer **Blue Cross Blue Shield of IL, MT,
NM, OK & TX**
Chicago, IL
Oct, 04

Web Developer - Entry Level **Verizon**
Temple Terrace, FL
Sep, 26

JAVA Developer **Advanced Technology Group
Inc**
Greenwood Village, CO
Oct, 06

1 2 ... 5913 »

☐ Freelance
 ☐ Full-time
 ☐ Internship
 ☐ Part-time
 ☐ All

Filter Results

jobs by **indeed**

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
03 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
04 <html xmlns="http://www.w3.org/1999/xhtml"
05       xmlns:h="http://java.sun.com/jsf/html"
06       xmlns:f="http://java.sun.com/jsf/core" >
07
08   <h:body>
09     <h1>JSF 2.2 Multi-Selection ListBox Example</h1>
10
11     <h:form>
12       1. Hardcoded with "f:selectItem" :
13       <br/>
14
15       <h:selectManyListbox value="#{user.favoriteCar1}">
16         <f:selectItem itemValue="520" itemLabel="BMW Series 5 - 520" />
17         <f:selectItem itemValue="525" itemLabel="BMW Series 5 - 525" />
18         <f:selectItem itemValue="535" itemLabel="BMW Series 5 - 535" />
19       </h:selectManyListbox>
20
21       <br/><br/>
22
23       2. Generated by Map:
24       <br/>
25
26       <h:selectManyListbox value="#{user.favoriteCar2}">
27         <f:selectItems value="#{user.favoriteCar2Value}" />
28       </h:selectManyListbox>
29
30       <br/><br/>
31
32       3. Generated by Object Array; access with "var"
33       <br/>
34       <h:selectManyListbox value="#{user.favoriteCar3}">
35         <f:selectItems value="#{user.favoriteCar3Value}" var="c"

```

GET THE JAVA SKILLS YOU NEED IN 2016.

Start Learning

```

40
41     <h:commandButton value="Submit" action="response"/>
42     <h:commandButton value="Reset" type="reset"/>
43   </h:form>
44 </h:body>
45 </html>

```

And the results page:

[response.xhtml](#)

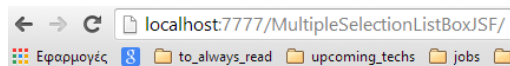
```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
03 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
04 <html xmlns="http://www.w3.org/1999/xhtml"
05       xmlns:h="http://java.sun.com/jsf/html">
06
07   <h:body>
08     <h1>JSF 2.2 Multiple-Selection ListBox Example - Response Page</h1>
09
10     <ol>
11       <li>user.favoriteCar1 : #{user.favoriteCar1InString}</li>
12       <li>user.favoriteCar2 : #{user.favoriteCar2InString}</li>
13       <li>user.favoriteCar3 : #{user.favoriteCar3InString}</li>
14     </ol>
15
16   </h:body>
17 </html>

```

3. Demo

I'll just make some selections:



JSF 2.2 Multi-Selection ListBo

1. Hardcoded with "f:selectItem" :

BMW Series 5 - 520
BMW Series 5 - 525
BMW Series 5 - 535

2. Generated by Map:

BMW Series 1 - 116
BMW Series 1 - 118
BMW Series 1 - 120

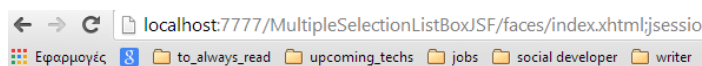
3. Generated by Object Array; access with "var"

BMW Series 3 - 316
BMW Series 3 - 318
BMW Series 3 - 320

Submit Reset



And here's what happened after submitting the selections:



JSF 2.2 Multiple-Selection ListBox Examn

JSF 2.0 Multiple Selection ListBox Example

1. `user.favoriteCar1 : [525]`
2. `user.favoriteCar2 : [118, 120]`
3. `user.favoriteCar3 : [316, 318, 320]`



This was an example of ListBox in JSF 2.0. You can also download the source code for this example: [MultipleSelectionListBoxJSF](#)

Tagged with: ECLIPSE

Do you want to know how to develop your skillset to become a **Java Rockstar?**

GET THE JAVA SKILLS YOU NEED IN 2016.

Start Learning

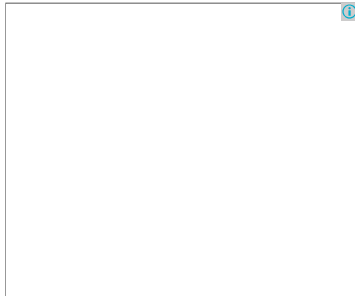
1. JPA Mini Book
2. JVM Troubleshooting Guide
3. JUnit Tutorial for Unit Testing
4. Java Annotations Tutorial
5. Java Interview Questions
6. Spring Interview Questions
7. Android UI Design

and many more

Email address:

Your email address

Sign up



KNOWLEDGE BASE

Courses

News

Resources

Tutorials

Whitepapers

THE CODE GEEKS NETWORK

.NET Code Geeks

Java Code Geeks

System Code Geeks

Web Code Geeks

HALL OF FAME

Android Alert Dialog Example

Android OnClickListener Example

How to convert Character to String and a String to Character Array in Java

Java Inheritance example

Java write to File Example

java.io.FileNotFoundException – How to solve File Not Found Exception

java.lang.arrayindexoutofboundsexception – How to handle Array Index Out Of Bounds Exception

java.lang.NoClassDefFoundError – How to solve No Class Def Found Error

JSON Example With Jersey + Jackson

Spring JdbcTemplate Example

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on creating the ultimate Java to Java developers resource center; targeted at the technical architect, technical team lead (senior developer), project manager and junior developers alike. JCGs serve the Java, SOA, Agile and Telecom communities with daily news written by domain experts, articles, tutorials, reviews, announcements, code snippets and open source projects.

DISCLAIMER

All trademarks and registered trademarks appearing on Java Code Geeks are the property of their respective owners. Java is a trademark or registered trademark of Oracle Corporation in the United States and other countries. Examples Java Code Geeks is not connected to Oracle Corporation and is not sponsored by Oracle Corporation.