



Mobile Benchmarks 2016

 Rate your app
performance
adjust.com


ANDROID ▾ CORE JAVA ▾ DESKTOP JAVA ▾ ENTERPRISE JAVA ▾ JAVA BASICS ▾ JVM LANGUAGES ▾ SOFTWARE DEVELOPMENT ▾ DEVOPS ▾

[Home](#) » [Enterprise Java](#) » [jsf](#) » [CheckBox 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 Intrasoft International S.A. He holds a diploma at Informatics & Telecommunications Engineering and is interested in continuous development.



CheckBox Example with JSF 2.0

Posted by: Thodoris Bais in jsf February 8th, 2014

Hi there, hope you had an interesting day. Today we're gonna talk about checkboxes in JSF 2.0. To represent a checkbox in JSF, we use the tag

```
h:selectBooleanCheckbox
```

. Ok, that's really easy, but what if we'd like to create a group of checkboxes, where the user could select more than one checkbox? This can be done using the

```
h:selectManyCheckbox
```

tag; the HTML renderings are exactly the same, as we saw in my previous example. To be more specific, before getting into the example's structure, here is a small example that implements a group of three checkboxes, where the user can select more than one of them:

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](#)

```
1 <h:selectManyCheckbox value="#{user.favNumber1}">
2   <f:selectItem itemValue="1" itemLabel="Number1 - 1" />
3   <f:selectItem itemValue="2" itemLabel="Number1 - 2" />
4   <f:selectItem itemValue="3" itemLabel="Number1 - 3" />
5 </h:selectManyCheckbox>
```

Ok, enough said, let's have a quick example with that type of checkboxes, but I have to first notice the four different ways that we can populate a group of checkboxes:

1. Hardcoded value in a

```
f:selectItem
```

tag.

2. Generated values from an Array and passed into the fore-mentioned tag.
3. Generated values using a Map and passed into the same tag.
4. 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

Here's the structure of the Bean that holds the submitted values.

```
UserBean.java
```

Jetta TSI
O esportivo disfarçado de sedan.

CLIQUE E CONHEÇA **Volkswagen**

Golf.
Com teto solar panorâmico.

CONHEÇA **Volkswagen**



JIRA:
My team runs on JIRA. Start a trial and get your shirt.

ad

NEWSLETTER

169,842 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!



```

001 package com.javacodegeeks.enterprise.jsf.checkboxes;
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 = -3953324291794510390L;
016
017     public boolean rememberMe;
018     public String[] favoriteCar1;
019     public String[] favoriteCar2;
020     public String[] favoriteCar3;
021     public String[] favoriteCar4;
022     public boolean isRememberMe() {
023         return rememberMe;
024     }
025     public void setRememberMe(boolean rememberMe) {
026         this.rememberMe = rememberMe;
027     }
028     public String[] getFavoriteCar1() {
029         return favoriteCar1;
030     }
031     public void setFavoriteCar1(String[] favoriteCar1) {
032         this.favoriteCar1 = favoriteCar1;
033     }
034     public String[] getFavoriteCar2() {
035         return favoriteCar2;
036     }
037     public void setFavoriteCar2(String[] favoriteCar2) {
038         this.favoriteCar2 = favoriteCar2;
039     }
040     public String[] getFavoriteCar3() {
041         return favoriteCar3;
042     }
043     public void setFavoriteCar3(String[] favoriteCar3) {
044         this.favoriteCar3 = favoriteCar3;
045     }
046     public String[] getFavoriteCar4() {
047         return favoriteCar4;
048     }
049     public void setFavoriteCar4(String[] favoriteCar4) {
050         this.favoriteCar4 = favoriteCar4;
051     }
052
053
054     public String getFavoriteCar1InString()
055     {
056         return Arrays.toString(favoriteCar1);
057     }
058
059     //generated by Array
060     public String[] getFavoriteCar2Value()
061     {
062         favoriteCar2 = new String [5];
063         favoriteCar2[0] = "116";
064         favoriteCar2[1] = "118";
065         favoriteCar2[2] = "X1";
066         favoriteCar2[3] = "Series 1 Coupe";
067         favoriteCar2[4] = "120";
068
069         return favoriteCar2;
070     }
071
072     public String getFavoriteCar2InString()
073     {
074         return Arrays.toString(favoriteCar2);
075     }
076
077     //generated by Map
078     private static Map<String, Object> car3Value;
079     static
080     {
081         car3Value = new LinkedHashMap<String, Object>();
082         car3Value.put("Car3 - 316", "BMW 316");
083         car3Value.put("Car3 - 318", "BMW 318");
084         car3Value.put("Car3 - 320", "BMW 320");
085         car3Value.put("Car3 - 325", "BMW 325");
086         car3Value.put("Car3 - 330", "BMW 330");
087     }
088
089     public Map<String, Object> getFavoriteCar3Value()
090     {
091         return car3Value;
092     }
093     public String getFavoriteCar3InString() {
094         return Arrays.toString(favoriteCar3);
095     }
096
097     //generated by Object Array
098     public static class Car
099     {
100         public String carLabel;
101         public String carValue;
102
103         public Car(String carLabel, String carValue)
104         {
105             this.carLabel = carLabel;

```

GET THE JAVA SKILLS YOU NEED IN 2016.

Start Learning

```

110     {
111         return carLabel;
112     }
113     public String getCarValue()
114     {
115         return carValue;
116     }
117 }
118
119 public Car[] car4List;
120 public Car[] getFavoriteCar4Value()
121 {
122     car4List = new Car[5];
123

```



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**

```

124         car4List = new Car[5];
125         car4List[0] = new Car("Car 4 - M3", "BMW M3 SMG");
126         car4List[1] = new Car("Car 4 - X3", "BMW X3");
127         car4List[2] = new Car("Car 4 - X5", "BMW X5");
128         car4List[3] = new Car("Car 4 - X6", "BMW X6");
129         car4List[4] = new Car("Car 4 - 745", "BMW 745");
130
131         return car4List;
132     }
133
134     public String getFavoriteCar4InString()
135     {
136         return Arrays.toString(favoriteCar4);
137     }
138 }

```

2. Our JSF Pages

First, the welcome page, where we have a single checkbox and the four afore-mentioned different ways, which populate group checkboxes.

index.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       xmlns:f="http://java.sun.com/jsf/core" >
07
08     <h:body>
09         <h1>JSF 2.2 CheckBoxes Example</h1>
10         <h:form>
11             <h2>1. Single checkbox</h2>
12             <h:selectBooleanCheckbox value="#{user.rememberMe}" />Remember Me
13
14             <h2>2. Group of checkboxes</h2>
15             1. Hardcoded using the tag "f:selectItem" :
16             <h:selectManyCheckbox value="#{user.favoriteCar1}">
17                 <f:selectItem itemLabel="Car1 - E10" itemValue="BMW E10" />
18                 <f:selectItem itemLabel="Car1 - E36" itemValue="BMW E36" />
19                 <f:selectItem itemLabel="Car1 - E46" itemValue="BMW E46" />
20                 <f:selectItem itemLabel="Car1 - E87" itemValue="BMW E87" />
21                 <f:selectItem itemLabel="Car1 - E92" itemValue="BMW E92" />
22             </h:selectManyCheckbox>
23
24             <br/>
25
26             2. Generated by Array :
27             <h:selectManyCheckbox value="#{user.favoriteCar2}">
28                 <f:selectItems value="#{user.favoriteCar2Value}" />
29             </h:selectManyCheckbox>
30
31             <br/>
32
33             3. Generated by Map :
34             <h:selectManyCheckbox value="#{user.favoriteCar3}">
35                 <f:selectItems value="#{user.favoriteCar3Value}" />
36             </h:selectManyCheckbox>
37
38             <br/>
39
40             4. Generated by Object, displayed using var
41             <h:selectManyCheckbox value="#{user.favoriteCar4}">
42                 <f:selectItems value="#{user.favoriteCar4Value}" var="last"
43                     itemLabel="#{last.carLabel}" itemValue="#{last.carValue}" />
44             </h:selectManyCheckbox>
45
46             <br/>
47             <h:commandButton value="Submit" action="results" />
48             <h:commandButton value="Reset" type="reset" />
49
50         </h:form>
51     </h:body>
52 </html>

```

Then, just to ensure that every submitted value saved correctly, we'll try to access the related

getters

through a JSF page:

results.xhtml

```

05       xmlns:f="http://java.sun.com/jsf/core" >
06
07     <h:body>
08         <h1>JSF 2.2 CheckBoxes Example - Response Page</h1>
09
10         <ol>
11             <li>user.rememberMe : #{user.rememberMe}</li>
12             <li>user.favoriteCar1 : #{user.favoriteCar1InString}</li>
13             <li>user.favoriteCar2 : #{user.favoriteCar2InString}</li>
14             <li>user.favoriteCar3 : #{user.favoriteCar3InString}</li>
15             <li>user.favoriteCar4 : #{user.favoriteCar4InString}</li>
16         </ol>
17     </h:body>
18 </html>

```

3. Demo

I'll just select my favorites from each group:

localhost:7777/CheckBoxesJSF/

Εμφανίσεις to_always_read upcoming_techs jobs social developer writer

JSF 2.2 CheckBoxes Example

1. Single checkbox

☒ Remember Me

2. Group of checkboxes

1. Hardcoded using the tag "fselectItem" :

☐ Car1 - E10 ☐ Car1 - E36 ☒ Car1 - E46 ☐ Car1 - E87 ☐ Car1 - E92

2. Generated by Array :

☐ 116 ☐ 118 ☐ X1 ☒ Series 1 Coupe ☐ 120

3. Generated by Map :

☒ Car3 - 316 ☒ Car3 - 318 ☐ Car3 - 320 ☐ Car3 - 325 ☒ Car3 - 330

4. Generated by Object, displayed using var

☒ Car 4 - M3 ☐ Car 4 - X3 ☐ Car 4 - X5 ☐ Car 4 - X6 ☐ Car 4 - 745



Let's see what happened:



JSF 2.2 CheckBoxes Example - Response Page

1. user.rememberMe : true
2. user.favoriteCar1 : [BMW E46]
3. user.favoriteCar2 : [Series 1 Coupe]
4. user.favoriteCar3 : [BMW 316, BMW 318, BMW 330]
5. user.favoriteCar4 : [BMW M3 SMG]



This was an example of CheckBoxes in JSF 2.0. You can also download the source code for this example: CheckBoxesJSF

Tagged with:

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

Subscribe to our newsletter to start Rocking **right now!**
To get you started we give you our best selling eBooks for **FREE!**

1. JPA Mini Book
2. JPA Mini Book
3. Java Interview Questions
4. Spring Interview Questions
5. Android UI Design
6. Spring Interview Questions
7. Android UI Design

and many more

Email address:

GET THE JAVA SKILLS YOU NEED IN 2016.

| |
|-------------|
| News |
| Resources |
| Tutorials |
| Whitepapers |

THE CODE GEEKS NETWORK

| |
|-------------------|
| .NET Code Geeks |
| Java Code Geeks |
| System Code Geeks |
| Web Code Geeks |

| |
|--|
| 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 |

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.

GET THE JAVA SKILLS YOU NEED IN 2016.

Start Learning