

## MultiBox : Insieme di checkbox in Struts

Questo tips & trick illustra come gestire, in maniera semplice, le checkbox tramite il framework Struts.

Ipotizziamo di trovarci nella situazione di dover visualizzare una serie di elementi, e tramite un form selezionarne alcune tramite l'uso di checkbox.

Molto probabilmente il Form Bean sarà composto da un ArrayList, dove ogni elemento rappresenterà i dati da visualizzare.

Bean : (chiamato **MovimentiBean**)

```
String codice;  
String nome;  
String numero_carta;  
Long importo;
```

Form Bean: (nello struts-config.xml definito come **movimentiForm**)

```
ArrayList movimenti;  
Object[] movimentiSelezionati;
```

Notare la presenza dell'array movimentiSelezionati, che verrà appunto utilizzato per memorizzare i movimenti selezionati dall'utente finale.

La pagina JSP conterrà:

```
<logic:iterate id="mov" name="movimentiForm" property=" movimenti" />  
  <tr>  
    <td><bean:write name="mov" property="nome" /></td>  
    <td><bean:write name="mov" property=" numero_carta" /></td>  
    <td><bean:write name="mov" property=" importo" /></td>  
    <td>  
      <html:multibox property="movimentiSelezionati">  
        <bean:write name="mov" property=" codice" />  
      </html:multibox>  
    </td>  
  </tr>  
</logic:iterate>
```

L'esempio qui sopra riportato visualizzerà per ogni riga, un diverso elemento dell'ArrayList presente nel formBean.

Il tag multibox indica la presenza di una serie di checkbox, il cui "risultato" (cioè l'elenco di quelle selezionate) sarà memorizzato all'interno dell'array **movimentiSelezionati**.

Tra i tag <html:multibox> e </html:multibox>, viene definito il "value" della checkbox, in questo particolare esempio viene utilizzato il codice del movimento, *in quanto univoco*. (*è importante che il valore sia diverso per ogni elemento checkbox*)

**NOTA:** All'interno di **movimentiSelezionati** verranno memorizzati proprio i value selezionati, cioè i codici dei movimenti "checkati".

## SE VOLESSI PRE-SELEZIONARE ALCUNE CHECKBOX?

L'esempio sopra presentato visualizzerà una maschera contenente una serie di checkbox, inizialmente non selezionate.

Ma se per qualche motivo la vostra applicazione presenta la necessità di inizializzare alcuni elementi già “checkati” e altri no?

Sempre grazie all'oggetto **movimentiSelezionati** è possibile inizializzare l'array di oggetti (in realtà Stringhe) contenente i “codici” di quelli che si desidera vengano preselezionati.

**ESEMPIO:** Ipotizziamo di voler pre-selezionare i codici relativi alle corte di credito, con movimenti che hanno un importo superiore a 10.000€:

Nel form bean, è possibile implementare il metodo di getter (**getMovimentiSelezionati**) nella seguente maniera:

```
ArrayList movimenti;  
Object[] movimentiSelezionati;  
  
public Object[]getMovimentiSelezionati () {  
    ArrayList preSelezionatiLista = new ArrayList();  
    MovimentiBean temp = new MovimentiBean ();  
    for (int i = 0; i< movimenti.size(); i++){  
        temp = (MovimentiBean) movimenti.get(i);  
        if (temp.getImporto() > 10000 )  
            preSelezionatiLista.add(temp.getCodice());  
    }  
    return preSelezionatiLista.toArray();  
}
```