

Struts2 iterator タグ

Iterator は、java.util.Collection、 java.util.Iterator を実装した値の繰り返し処理を行う。

タグ名	必須	デフォルト	評価	型	内容
id	false		true	String	要素参照用 ID. UI や、FORM タグ用。HTML の id 属性として利用される。
status	false	false	true	Boolean	指定された場合、IteratorStatus のインスタンスは、スタックの上に積まれる。
value	false		true	String	繰り返し可能なソース、その他、新たに生成されたリストに自身が登録されたオブジェクト。

例

value stack 上のカレントオブジェクトの getDays() メソッドの値を参照し、繰り返し利用する例。

<s:property/> タグがイテレータの現在値を出力する。

```
<s:iterator value="days">
  <p>day is: <s:property/></p>
</s:iterator>
```

Bean タグを利用し、ActionContext に格納する例。iterator タグは、ActionContext からオブジェクトを参照し、getDays() メソッドを呼び出す。status 属性も、IteratorStatus オブジェクトを生成するのに使用され、例では、odd() メソッドが、行の色を変更するのに利用されている。

```
<s:bean name="org.apache.struts2.example.IteratorExample" id="it">
  <s:param name="day" value="'foo'"/>
  <s:param name="day" value="'bar'"/>
</s:bean>
<p/>
<table border="0" cellspacing="0" cellpadding="1">
  <tr>
    <th>Days of the week</th>
  </tr>
</table>
<p/>
<s:iterator value="#it.days" status="rowstatus">
  <tr>
    <s:if test="#rowstatus.odd == true">
```

```

        <td style="background: grey"><s:property/></td>
    </s:if>
    <s:else>
        <td><s:property/></td>
    </s:else>
</tr>
</s:iterator>
</table>

```

次の例は、かなり status 属性を実例的に説明している。DAO を action クラスから (Object-Graph Navigation Language 式言語の一種。JSP や JSF の EL と比較して非常に強力な記述能力を持つ) 経由で使用する場合、グループおよび所属するユーザを繰り返し処理する。last() メソッドは、カレントオブジェクトが最後の要素であることを示し、最終要素でない場合、ユーザをカンマで区切る。

```

<s:iterator value="groupDao.groups" status="groupStatus">
    <tr class="<s:if test="#groupStatus.odd == true ">odd</s:if><s:else>even</s:else>">
        <td><s:property value="name" /></td>
        <td><s:property value="description" /></td>
        <td>
            <s:iterator value="users" status="userStatus">
                <s:property value="fullName" /><s:if test="!#userStatus.last">,</s:if>
            </s:iterator>
        </td>
    </tr>
</s:iterator>

```

action のコレクションを繰り返し処理し、イテレータの値を他の action に渡す。ここでは、'[0]' というトリックが使われている。これは、

カレントのイテレータの値を取得し、edit action に引き渡す。'[0]' を利用すると、<s:property /> と同様の効果が得られる。

```

<s:action name="entries" id="entries"/>
    <s:iterator value="#entries.entries" >
        <s:property value="name" />
        <s:property />
        <s:push value="...">
            <s:action name="edit" id="edit" >
                <s:param name="entry" value="[0]" />
            </s:action>
        </push>
    </s:iterator>

```

単純なループを iterator タグでシミュレートするには、以下のようにします。

```

<s:iterator status="stat" value="{1,2,3,4,5}" >
    <!-- grab the index (start with 0 ... ) -->
    <s:property value="#stat.index" />

    <!-- grab the top of the stack which should be the -->
    <!-- current iteration value (0, 1, ... 5) -->
    <s:property value="top" />
</s:iterator>

```