

STRUTS 2 - THE ITERATOR TAGS

http://www.tutorialspoint.com/struts_2/struts_iterator_tags.htm

Copyright © tutorialspoint.com

Create action classes

First of all let us create a simple class called Employee.java which looks like:

```
package com.tutorialspoint.struts2;

import java.util.ArrayList;
import java.util.List;

import org.apache.struts2.util.SubsetIteratorFilter.Decider;

public class Employee {
    private String name;
    private String department;

    public Employee(){}
    public Employee(String name,String department)
    {
        this.name = name;
        this.department = department;
    }
    private List employees;
    private List contractors;

    public String execute() {
        employees = new ArrayList();
        employees.add(new Employee("George","Recruitment"));
        employees.add(new Employee("Danielle","Accounts"));
        employees.add(new Employee("Melissa","Recruitment"));
        employees.add(new Employee("Rose","Accounts"));

        contractors = new ArrayList();
        contractors.add(new Employee("Mindy","Database"));
        contractors.add(new Employee("Vanessa","Network"));
        return "success";
    }

    public Decider getRecruitmentDecider() {
        return new Decider() {
            public boolean decide(Object element) throws Exception {
                Employee employee = (Employee)element;
                return employee.getDepartment().equals("Recruitment");
            }
        };
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDepartment() {
        return department;
    }
    public void setDepartment(String department) {
        this.department = department;
    }
    public List getEmployees() {
        return employees;
    }
    public void setEmployees(List employees) {
        this.employees = employees;
    }
}
```

```

    public List getContractors() {
        return contractors;
    }
    public void setContractors(List contractors) {
        this.contractors = contractors;
    }
}

```

The Employee class has two attributes - **name** and **department**, we also have two lists of employees - the permanent **employees** and the **contractors**. We have a method called **getRecruitmentDecider** that returns a **Decider** object. The Decider implementation returns **true** if the employee works for the **recruitment** department, and it returns **false** otherwise.

Next, let us create a **DepartmentComparator** to compare Employee objects:

```

package com.tutorialspoint.struts2;

import java.util.Comparator;

public class DepartmentComparator implements Comparator {
    public int compare(Employee e1, Employee e2) {
        return e1.getDepartment().compareTo(e2.getDepartment());
    }

    @Override
    public int compare(Object arg0, Object arg1) {
        return 0;
    }
}

```

As shown in the above example, the department comparator compares the employees based on the department in alphabetical order.

Create views

Create a file called **employee.jsp** with the following contents:

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
<title>Employees</title>
</head>
<body>
<b>Example of Iterator Tag</b><br/>
<s:iterator value="employees">
    <s:property value="name"/> ,
    <s:property value="department"/><br/>
</s:iterator>
<br/><br/>
<b>Employees sorted by Department</b><br/>

<s:bean name="com.tutorialspoint.struts2.DepartmentComparator"
    var="deptComparator" />

<s:sort comparator="deptComparator" source="employees">
    <s:iterator>
        <s:property value="name"/> ,
        <s:property value="department"/><br/>
    </s:iterator>
</s:sort>
<br/><br/>
<b>SubSet Tag - Employees working in Recruitment department </b><br/>
<s:subset decider="recruitmentDecider" source="employees">
    <s:iterator>
        <s:property value="name"/> ,
        <s:property value="department"/><br/>

```

```

    </s:iterator>
</s:subset>
<br/><br/>
<b>SubSet Tag - Employees 2 and 3 </b><br/>
<s:subset start="1" count="2" source="employees">
    <s:iterator>
        <s:property value="name"/> ,
        <s:property value="department"/><br/>
    </s:iterator>
</s:subset>
</body>
</html>

```

Let us go through the used tags one by one:

Iterator tag

We are using the **iterator** tag to go through the employees list. We supply the "employees" property as the source to the iterator tag. In the body of the iterator tag, we now have access to the Employee object in the employees list. We print the name of the employee followed by their department.

Sort tag

First of all we declared the **DepartmentComparator** as a bean. We gave this bean a name **deptComparator**. Then we used the **sort** tag and specify the "employees" list as the source and the "deptComparator" as the comparator to use. Then, as per the previous example, we iterate the list and print the employees. As you can see from the output, this prints the list of employees sorted by department

Subset tag

The **subset** tag is used to get a sub set of the list or array. We have two flavours of subset tag. In the first example, we use the **recruitmentDecider** to get the list of employees who work for the recruitment department *please see the getRecruitmentDecider() method in Employee.java*.

In the second example, we are not using any deciders but instead we are after elements 2 and 3 in the list. The subset tag takes in two parameters "count" and "start". "start" determines the starting point of the subset and the "count" determines the length of the subset.

Configuration Files

Your **struts.xml** should look like:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
    <constant name="struts.devMode" value="true" />

    <package name="helloworld" extends="struts-default">
        <action name="employee"

            method="execute">
                <result name="success">/employee.jsp</result>
            </action>
        </package>
    </struts>

```

Your **web.xml** should look like:

```

<?xml version="1.0" encoding="UTF-8"?>

```

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  >

  <display-name>Struts 2</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>

  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

Right click on the project name and click **Export > WAR File** to create a War file. Then deploy this WAR in the Tomcat's webapps directory. Finally, start Tomcat server and try to access URL <http://localhost:8080/HelloWorldStruts2/employee.action>. This will give you following screen:

Loading [MathJax]/jax/output/HTML-CSS/jax.js