STRUTS FRAMEWORK COURSE

MESSAGES IN STRUTS 2 FRAMEWORK

Eng. Ubaldo Acosta

By the expert: Ing. Ubaldo Acosta

Global Mentoring

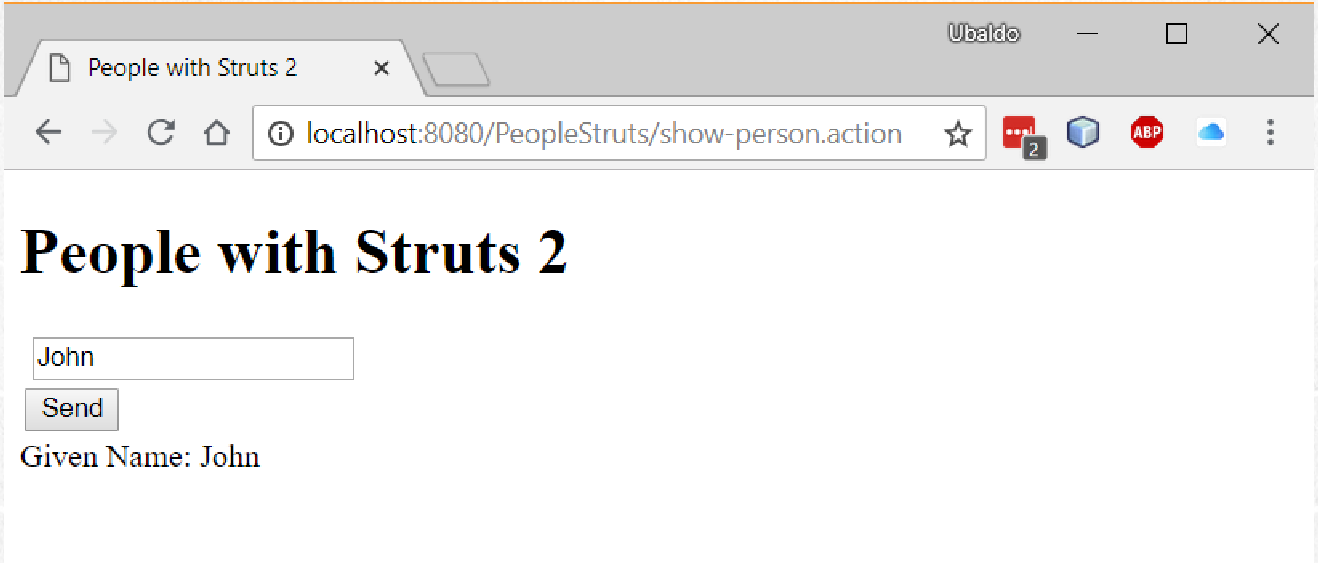
JAVA ONLINE COURSES UNIVERSITY

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create an application to implement the use of Messages with Struts 2 Framework. At the end we should observe the following:



People with Struts 2

John

Send

Given Name: John

EXERCISE REQUIREMENT

We are going to start from the previous exercise called PeopleStruts.

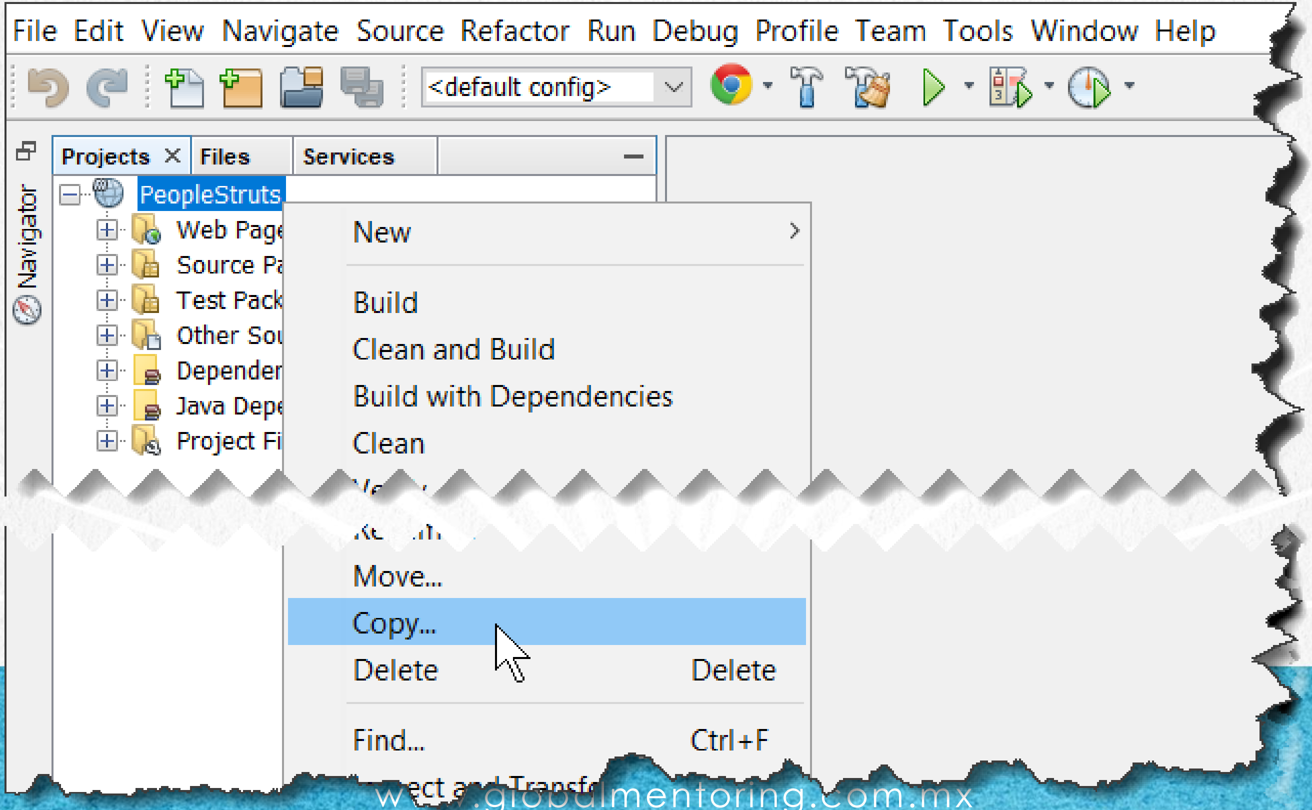
In this exercise we will handle each of the labels (labels) of the HTML page as messages in a separate property file.

This is to apply the concept of internationalization later on, where we can add several languages to our Web application with Struts 2. But at the moment, we are going to add the concept of messages or labels from a property file.

Global Mentoring

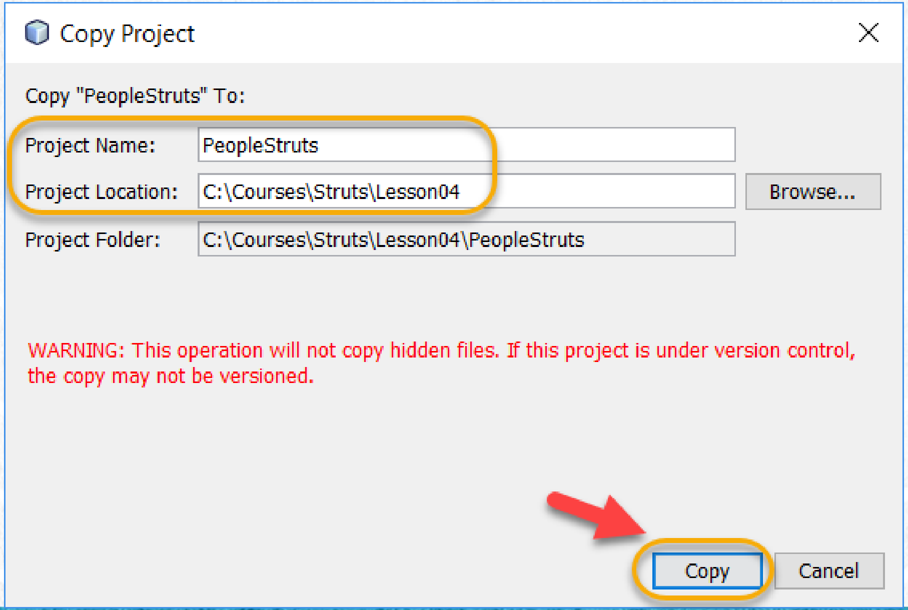
1. COPY THE PROJECT

- We created the new project using the copy function on the previous project:



1. COPY THE PROJECT

- We write the following values as shown and click on copy. This will not change the name of the Maven project, so we'll leave it:



Copy Project

Copy "PeopleStruts" To:

Project Name: PeopleStruts

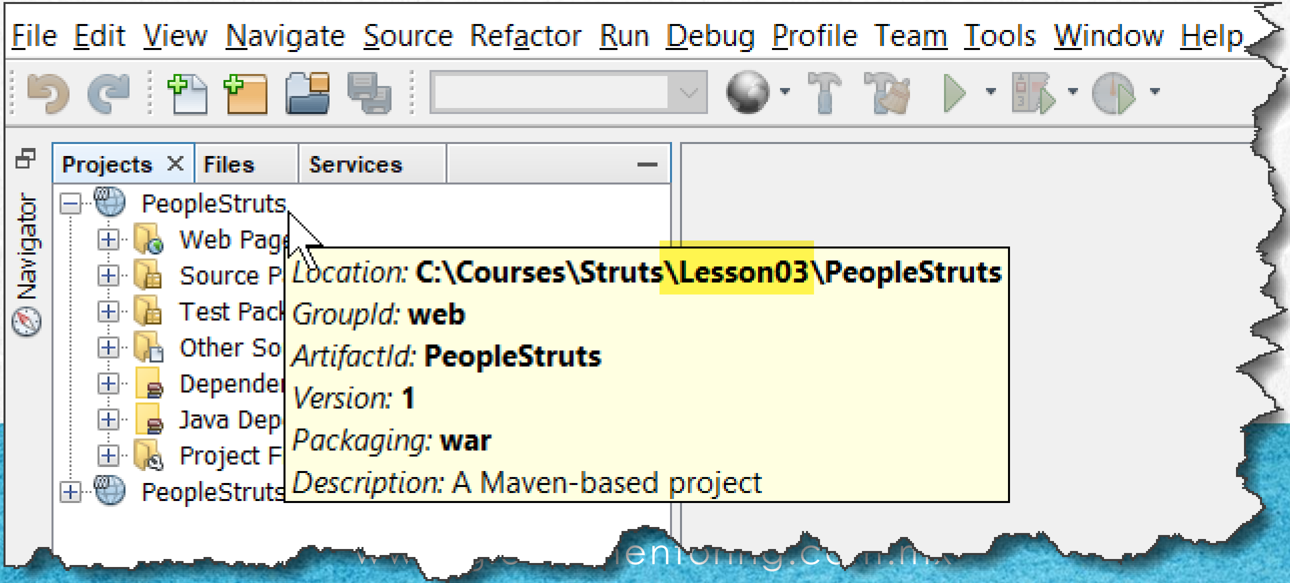
Project Location: C:\Courses\Struts\Lesson04

Project Folder: C:\Courses\Struts\Lesson04\PeopleStruts

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

2. CLOSE THE PROJECT THAT WE NO LONGER USE

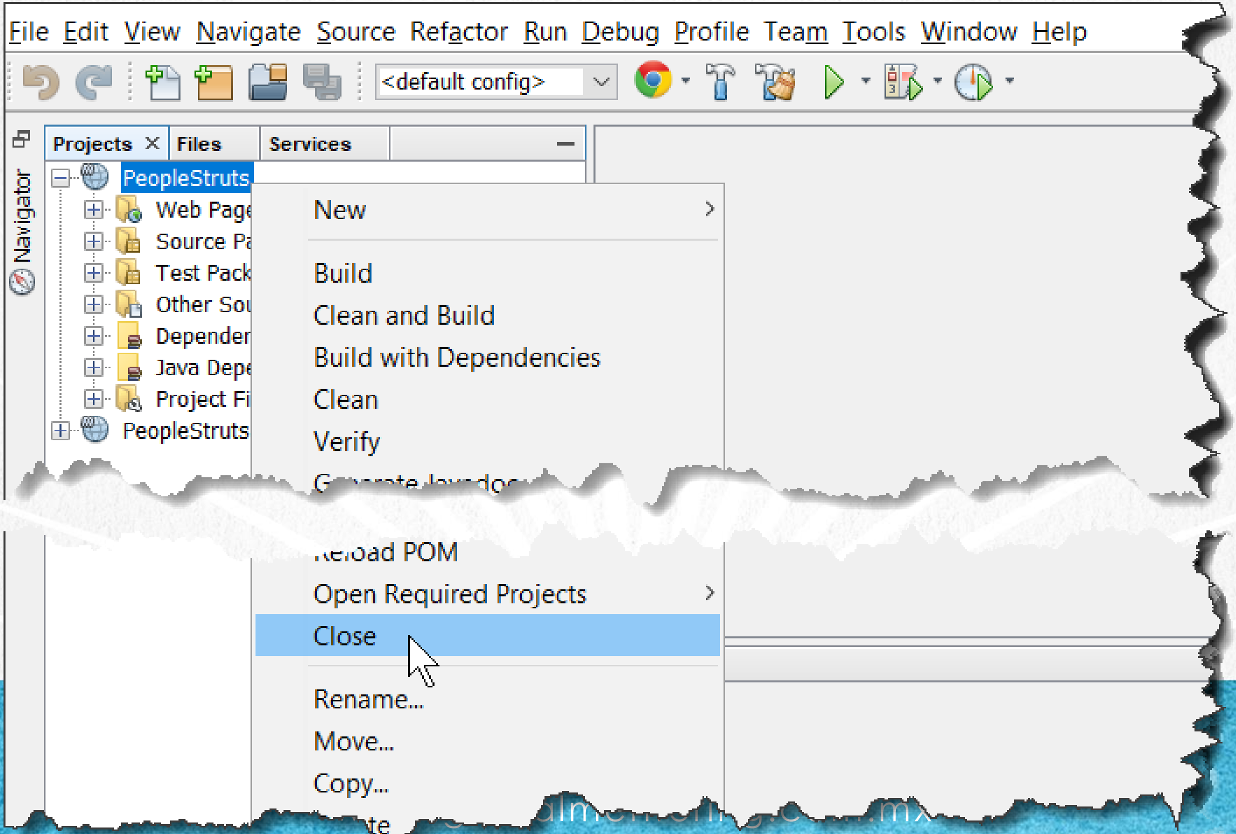
- We close the project that we no longer use, in this case to know which is the project that we must close (Lesson03), we can position ourselves on the name of the project and it will give us more information, including the path where the file is located. So we will know which project to close and which project to leave open :



Location: **C:\Courses\Struts\Lesson03\PeopleStruts**
GroupId: **web**
ArtifactId: **PeopleStruts**
Version: **1**
Packaging: **war**
Description: A Maven-based project

2. CLOSE THE PROJECT THAT WE NO LONGER USE

- We close the project that we no longer use.



3. CREATE A PROPERTY FILE

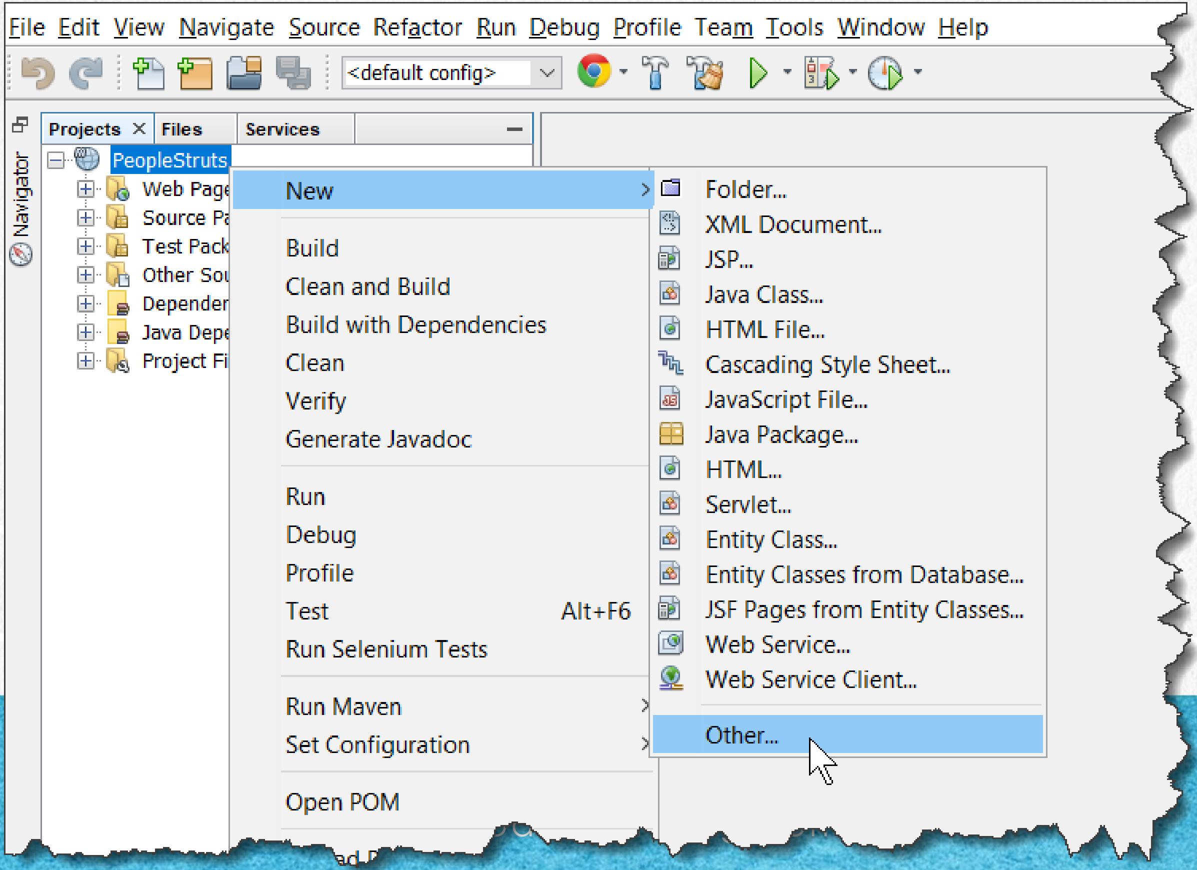
We are going to create next a file of properties that will contain the messages that we are going to handle for our Web application.

This file will continue to handle the configuration by conventions, so it must be called equal to the Action class that we have created. Therefore the file will be called: ShowPersonAction.properties

Because we are using Maven, this file must be deposited in the resources folder in a package identical to where the Action class is located, let's see how:

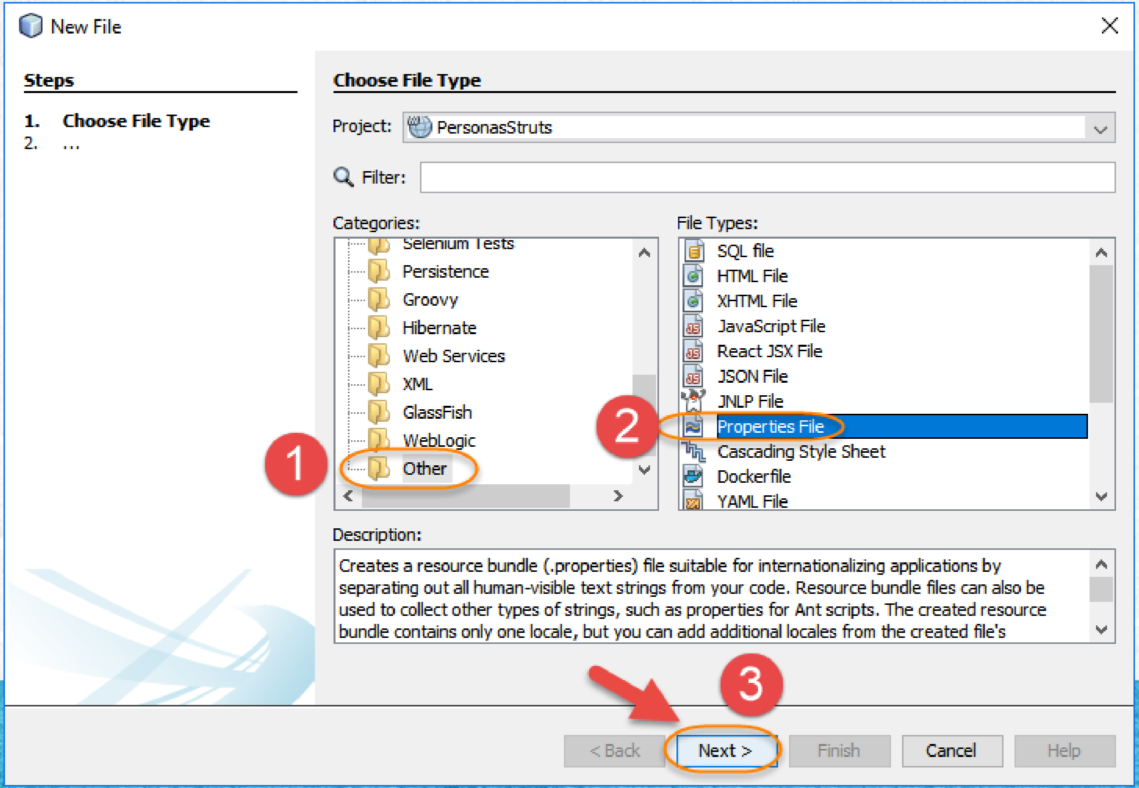
3 . CREATE A PROPERTIES FILE

- We select the option of another to find the type of property file:



3 . CREATE A PROPERTIES FILE

- Select the Other option, then Properties File and click Next :



New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: PersonasStruts

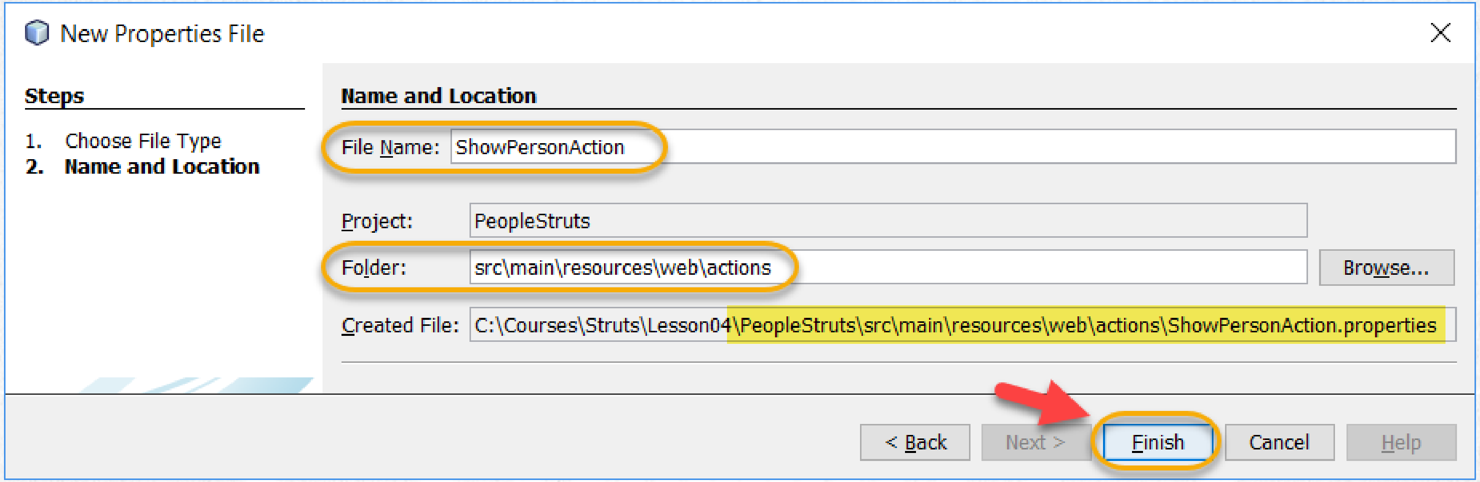
Filter:

Description:

Creates a resource bundle (.properties) file suitable for internationalizing applications by separating out all human-visible text strings from your code. Resource bundle files can also be used to collect other types of strings, such as properties for Ant scripts. The created resource bundle contains only one locale, but you can add additional locales from the created file's

3 . CREATE A PROPERTIES FILE

- We provide the values as shown. The path is the following :



New Properties File

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: ShowPersonAction

Project: PeopleStruts

Folder: src\main\resources\web\actions

Created File: C:\Courses\Struts\Lesson04\PeopleStruts\src\main\resources\web\actions\ShowPersonAction.properties

4. MODIFY THE CODE

ShowPersonAction.properties:

Click to download

```
person.title: People with Struts 2
person.value: Given Name
person.button: Send
```

5. MODIFY THE ACTION CLASS

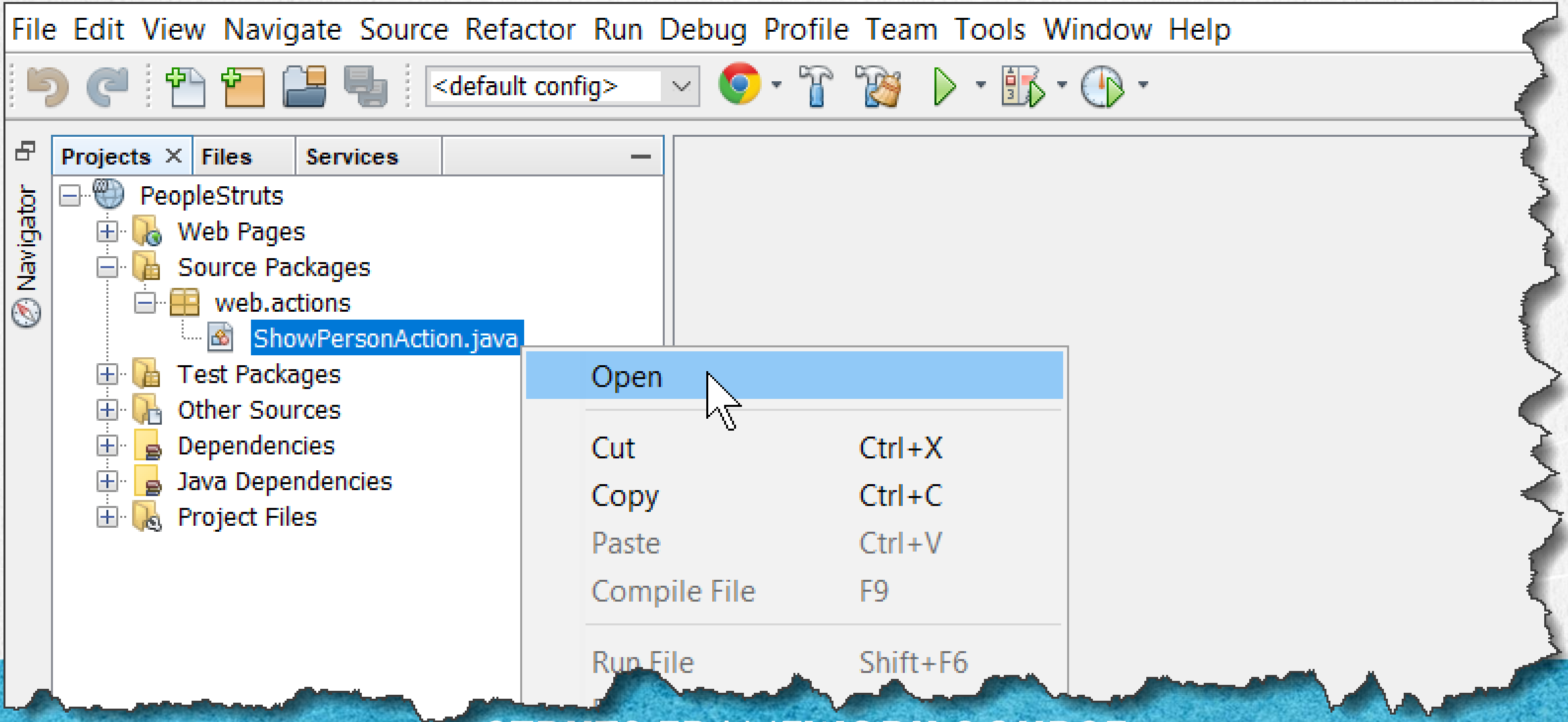
We modify the ShowPersonAction.java class to read the information of the created properties file. We must remember that we have inherited from the ActionSupport class, and therefore we have access to the getText method ("propertyName"), which is in charge of accessing the properties file and returning the value of the property name provided.

For each of the properties we will create a get method that corresponds to each property of the properties file, and this method will be used in turn in the JSP to access each of these properties. This is not always necessary, we will see that there are simpler ways to access the properties. This we will do later.

For the moment, we are going to modify the ShowPersonaAction.java class

5. MODIFY THE ACTION CLASS

- We open the ShowPersonAction.java class to modify it:



6. MODIFY THE CODE

ShowPersonAction.java:

Click to download

```
package web.actions;

import com.opensymphony.xwork2.ActionSupport;
import org.apache.logging.log4j.*;

public class ShowPersonAction extends ActionSupport {

    private String name;

    Logger log = LogManager.getLogger(ShowPersonAction.class);

    @Override
    public String execute() {
        log.info("The name is:" + this.name);
        return SUCCESS;
    }
```

6. MODIFY THE CODE

ShowPersonAction.java:

Click to download

```
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

      public String getTitle(){
        return getText("person.title");
    }

    public String getValue(){
         return getText("person.value");
    }

    public String getButton(){
         return getText("person.button");
    }
}
```

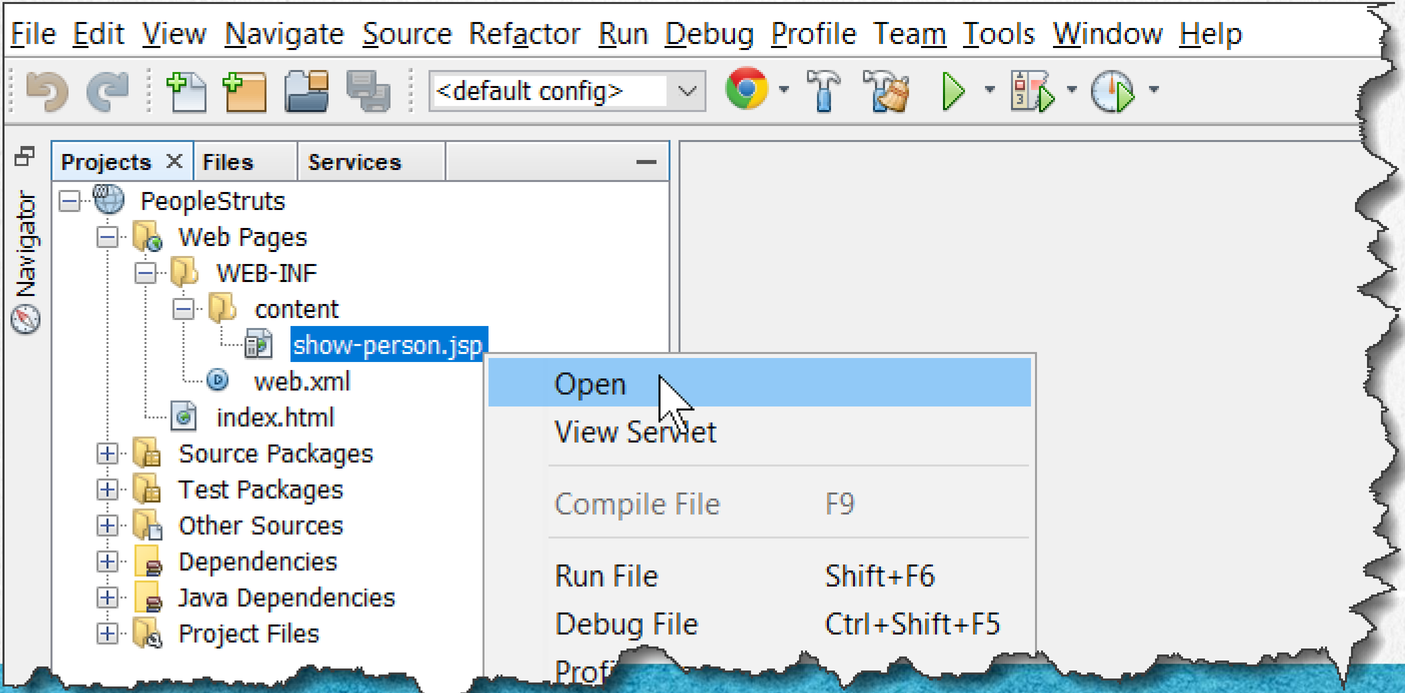
7. MODIFY THE JSP

For each of the properties we have created a get method that corresponds to each property of the properties file in the ShowPersonaAction.java class, and this method in turn will be used in the JSP to access each of these properties.

Let's see how to access each of these properties. We must remember that because we are using the JavaBeans nomenclature, it is not necessary to declare an attribute to access the respective get method. Only using the annotation of JavaBeans is it possible to access the get method as if it were a property.

7. MODIFY THE JSP

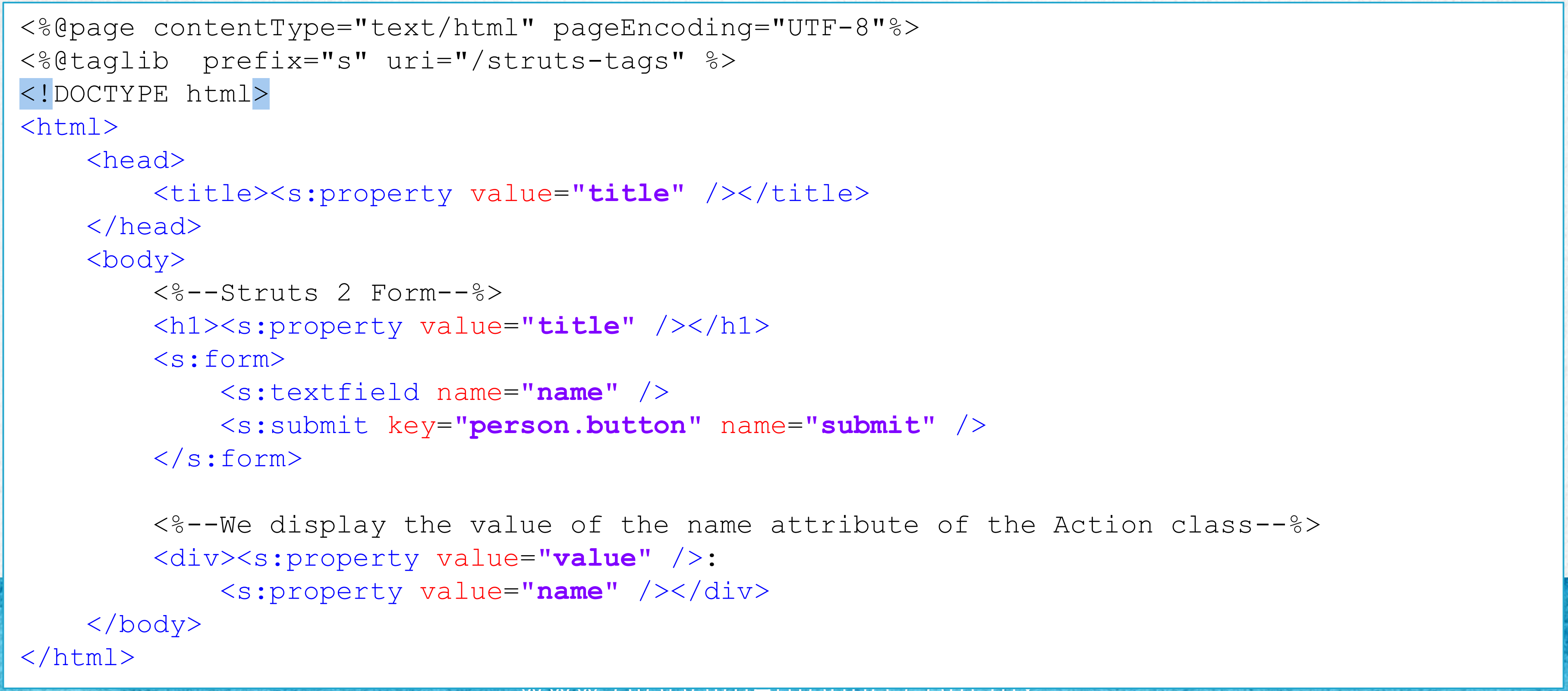
- We open the file show-person.jsp:



8. MODIFY THE CODE

show-person.jsp:

Click to download

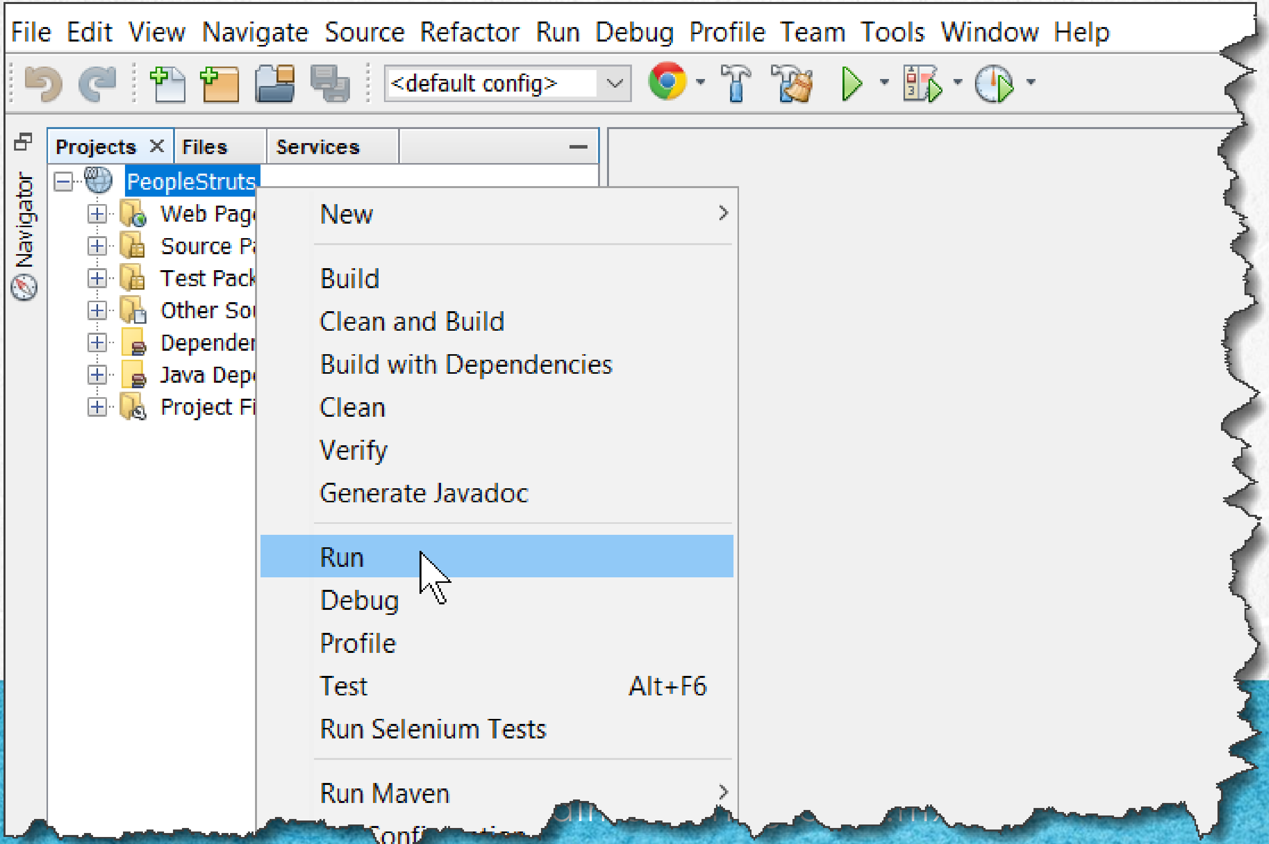


```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib  prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
    <head>
        <title><s:property value="title" /></title>
    </head>
    <body>
        <%--Struts 2 Form--%>
        <h1><s:property value="title" /></h1>
        <s:form>
            <s:textfield name="name" />
            <s:submit key="person.button" name="submit" />
        </s:form>

        <%--We display the value of the name attribute of the Action class--%>
        <div><s:property value="value" />:
            <s:property value="name" /></div>
    </body>
</html>
```

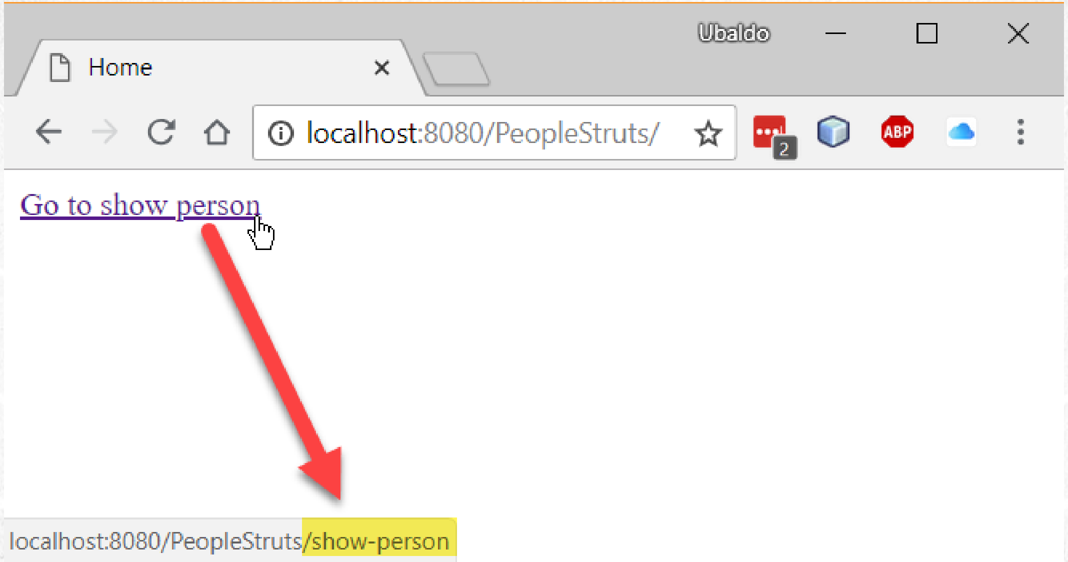
9. EXECUTE THE APPLICATION

- We execute the PeopleStruts application as follows :



9. EXECUTE THE APPLICATION

- We execute the application as follows:

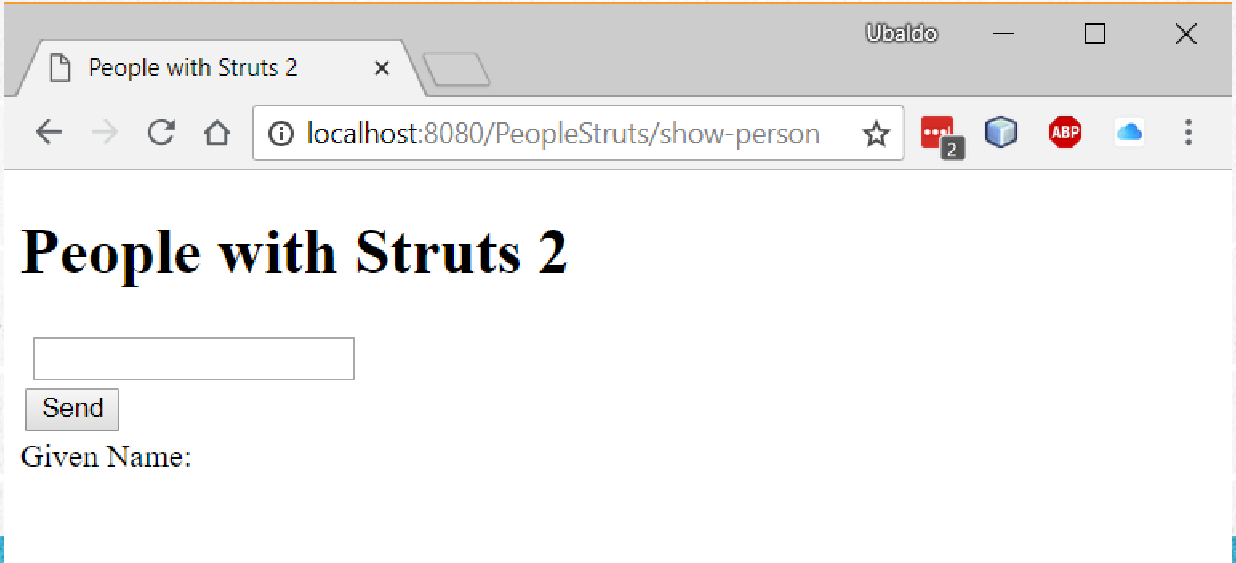


Go to show person

localhost:8080/PeopleStruts/show-person

9. EXECUTE THE APPLICATION

- We observe the form initially. No value is initially displayed in the provided name. But we note that it loads the messages that we have added to the properties file:



People with Struts 2

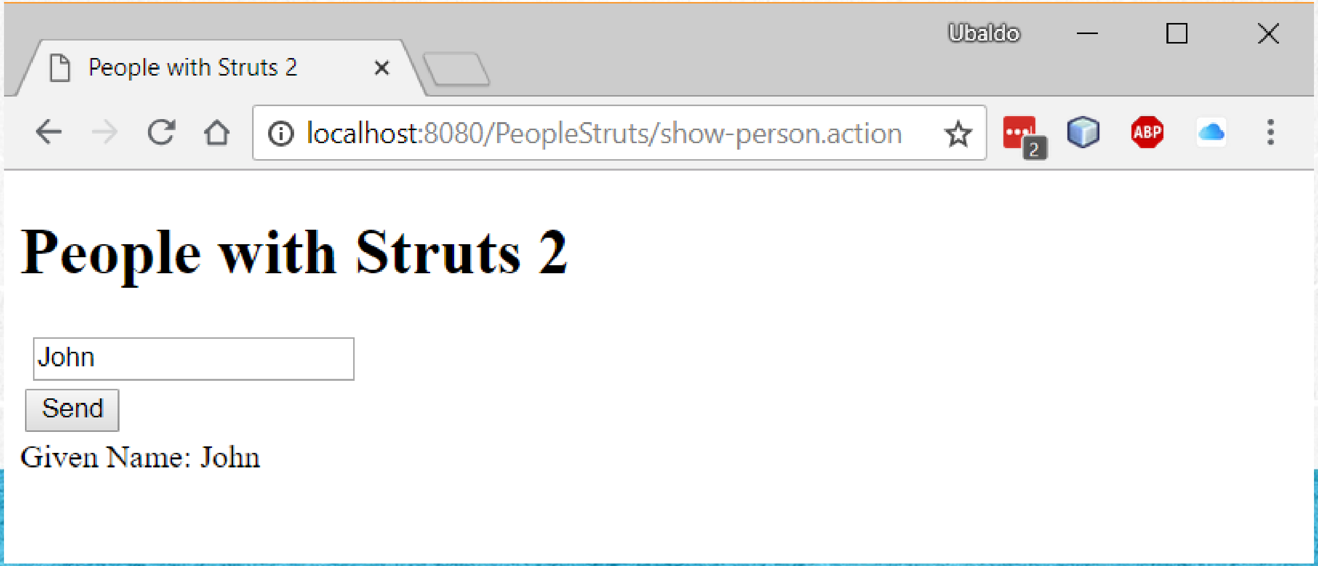
Send

Given Name:

9. Execute the Application

- We provide some value in the field of the form, eg: John, and send it to the server. Because we are using the Struts conventions, the URL that will be executed is: http://localhost:8080/PersonasStruts/mostrar-persona.action

- The extension .action that is added to the end is the default extension of Struts 2.



People with Struts 2

John

Send

Given Name: John

FINAL RECOMMENDATIONS

If for some reason the exercise fails, several things can be done to correct it:

1. Stop the Glassfish server
2. Make a Clean & Build project to have the most recent version compiled
3. Restart the project (deploy the project to the server again)

If the above does not work, you can try loading the resolved project which is 100% functional and rule out configuration problems in your environment or any other code error.

The configuration by conventions of Struts 2, is very sensitive, in such a way that everything must be written as it was specified in the exercise, since any change in the names will cause that the exercise is not executed correctly. However, the resolved project is 100% functional, so you can use these projects at any moment.

EXERCISE CONCLUSION

With this exercise we put into practice the handling of messages or labels in a resource bundle type file with Struts 2 (property file).

This is the basis for the concept of internationalization. Which we will see in the following exercise.

We observed that inheritance from the ActionSupport class already begins to show its advantages, since we very easily accessed the properties of the properties file and in this way the messages of the HTML page are already separated from the page and can be modified separately without need to modify the Java code or the JSP.

Global
Mentoring

Experiencia y Conocimiento para tu vida

CURSO ONLINE

STRUTS 2 FRAMEWORK

Por: Ing. Ubaldo Acosta

Global Mentoring

Experiencia y Conocimiento para tu vida

UNIVERSIDAD JAVA

CURSO ONLINE

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx