

STRUTS 2 - THE TEXT TAGS

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

Copyright © tutorialspoint.com

The **text** tag is a generic tag that is used to render a l18n text message. Follow one of the three steps:

- The message must be in a resource bundle with the same name as the action that it is associated with. In practice this means that you should create a properties file in the same package as your Java class with the same name as your class, but with .properties extension.
- If the named message is not found, then the body of the tag will be used as default message.
- If no body is used, then the name of the message will be used.

Let us check the following example to understand the usage of **text** tag:

Create action classes

```
package com.tutorialspoint.struts2;

public class HelloWorldAction{
    private String name;

    public String execute() throws Exception {
        return "success";
    }

    public String getName() {
        return name;
    }

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

Create views

Let us have **HelloWorld.jsp** with the following content:

```
<%@ taglib prefix="s" uri="/struts-tags"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Text Tag Example</title>
</head>
<body>

<s:i18n name="HelloWorldAction">
    <s:text name="name.success"/><br>
    <s:text name="name.xyz">Message doesn't exists</s:text><br>
    <s:text name="name.msg.param">
        <s:param >ZARA</s:param>
    </s:text>
</s:i18n>

</body>
</html>
```

Configuration Files

Let us create a property file with the same name as of your action class package name. So in this

case we will create **HelloWorldAction.properties** file and keep in the class path:

```
name.success = This is success message
name.msg.param = The param example - param : {0}
```

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" />
    <constant name="struts.custom.i18n.resources"
        value="ApplicationResources"/>

    <package name="helloaction" extends="struts-default">
        <action name="hello"

            method="execute">
                <result name="success">/HelloWorld.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/hello.action>. This will give you following screen: