

HelloWorld com Struts 2

Para esta pequena aplicação será utilizado como IDE o [eclipse](#) e a versão do [struts 2.0.11](#).

Do pacote **lib** do struts 2 utilizemos a seguintes bibliotecas:

- commons-logging.jar
- freemarker.jar
- ognl.jar
- struts2-core.jar
- xwork.jar

Essas libs deverão estar dentro de **WebContent/WEB-INF/lib**, esses diretórios são criados automaticamente quando se cria um WebDynamicProject no eclipse.

Primeiro será configurado o arquivo **web.xml**, serão adicionadas seguintes linhas:

```
<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>
```

Agora criaremos a nossa Action que vai se chamar **HelloStruts2**, ela será responsável por atribuir a string "José" para o atributo name da classe, ela deve estender de **ActionSupport** para que tenha as funcionalidades do Struts. O struts 2 usa os métodos *getter and setters* para acessar os atributos da classe. O método execute é o primeiro método que será chamado por padrão na action.

```

public class HelloStruts2 extends ActionSupport {
    private String _name;

    public HelloStruts2() {
    }

    public String execute(){
        _name = "José";
        return "execute";
    }

    public String getName(){
        return _name;
    }

    public void setName(String name){
        _name = name;
    }
}

```

Agora criaremos o arquivo **struts.xml**, que será reponsável pelo mapeamento da action, ele devera estar dentro da pasta src do projeto. na tag `<action>` colocamos o nome que será chamada a nossa action no atributo *name*, e o pacote em que a action se localiza no atributo *class*. Como não estamos especificando nenhum método, por default será chamado o método *execute*. Na tag `<result>` configuramos os rsultados que a action pode nos retornar. Neste caso estamos dizendo que a pode retornar a string *"execute"* e que se retornar a pagina direcionada será a **helloStruts.jsp**.

```

<?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>
    <include file="struts-default.xml" />

    <package name="default" extends="struts-default">
        <action name="HelloStruts"
            class="cris.helloworld.struts2.action.HelloStruts2">
            <result name="execute">helloStruts.jsp</result>
        </action>
    </package>
</struts>

```

Agora criaremos a nossa **index.jsp**, que será a primeira página da aplicação, nela colocamos a taglib do struts, por conversão usamos como prefixo a letra *"s"*, na tag `<a>` no atributo *href*, colocamos o nome da action, a qual foi mapeada no **struts.xml**. Então quando o link for clicado ele chamará o método **execute** da nossa action.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Index - Hello World Struts 2</title>
</head>
<body>
    <%@ taglib prefix="s" uri="/struts-tags"%>
    <s:a href="HelloStruts.action">Clique aqui!</s:a>
</body>
</html>

```

Por último será criado a página **helloStruts.jsp**, esta página será chamada quando o link da **index.jsp** for clicado. E a tag `<s:property>` será responsável por pegar o valor do atributo *name* atribuído na action:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Hello Struts 2</title>
</head>
<body>
    <%@ taglib prefix="s" uri="/struts-tags"%>
    Hello Struts 2 <s:property value="name"/>
</body>
</html>

```