Struts 2 Configuration File

1. Struts 2 Configuration File
2. Elements of struts.xml file

The struts application contains two main configuration files **struts.xml** file and **struts.properties** file.

The struts.properties file is used to override the default values of **default.xml** file provided by struts framework. So it is not mandatory. Mostly, you will not use struts.properties file. We will learn about it later.

Here, we are going to learn all about struts.xml file. First of all let us see the simple example of struts.xml file

struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
<package name="default" extends="struts-default">

<action name="product" class="com.javatpoint.Product">
<result name="success">welcome.jsp</result>
</action>

</package>
</struts>
```

1) package element

We can easily divide our struts application into sub modules. The package element specifies a module. You can have one or more packages in the struts.xml file.

Attributes of package element

- **name** name is must for defining any package.
- **namespace** It is an optional attribute of package. If namespace is not present, / is assumed as the default namespace. In such case, to invoke the action class, you need this URI:

1. /actionName.action
   If you specify any namespace, you need this URI:

2. /namespacename/actionName.action

- **extends** The package element mostly extends the **struts-default** package where interceptors and result types are defined. If you extend struts-default, all the actions of this package can use the interceptors and result-types defined in the **struts-default.xml** file.

2) action element

The action is the subelement of package and represents an action.

Attributes of action element

- **name** name is must for defining any action.
- **class** class is the optional attribute of action. If you omit the class attribute, **ActionSupport** will be considered as the default action. A simple action may be as:

1. <action name="product">

- **method** It is an optional attribute. If you don't specify method attribute, **execute** method will be considered as the method of action class. So this code:

1. <action name="product" **class**="com.javatpoint.Product">
   will be same as:

2. <action name="product" **class**="com.javatpoint.Product" method="execute">
   If you want to invoke a particular method of the action, you need to use method attribute.

3) result element

It is the sub element of action that specifies where to forward the request for this action.

Attributes of result element

- **name** is the optional attribute. If you omit the name attribute, success is assumed as the default result name.
- **type** is the optional attribute. If you omit the type attribute, dispatcher is assumed as the default result type.

The Struts Configuration File Tags

Table 18-1 lists and describes each of the tags used to configure the Struts configuration file.

Table 18-1: Struts Configuration File Tags

Tag	Description
action	Maps an application URL either to an **Action** object that will be executed when the specified URL is requested or to another URL that will be forwarded to.
action-mappings	Encapsulates the set of actions the application will have.
controller	Defines several global configuration settings for a Struts application.
exception	Defines an exception handler to process a specific exception thrown by an **Action**.
form-bean	Defines a Form Bean and assigns a logical name to it.
form-beans	Encapsulates the set of Form Beans the application will have.
form-property	Defines a form property for dynamic Form Beans.
forward	Defines a logical name for a URL, thus allowing code to reference the logical name and not the URL itself.

Table 18-1: Struts Configuration File Tags

Tag	Description
global-exceptions	Encapsulates a set of exception handlers, defined by **exception** tags, which are global to the application.
global-forwards	Encapsulates a set of forwards, defined by **forward** tags, which are global to the application.
message-resources	Defines a resource bundle that Struts will use when looking up externalized strings, messages, and labels.
plug-in	Defines a plugin that Struts loads at application startup and unloads at application shutdown.
set-property	Defines a property and its value.
struts-config	Is the root tag for the Struts configuration file and thus encapsulates all other tags in the file.

The Struts-config.xml File

The struts-config.xml configuration file is a link between the View and Model components in the Web Client but you would not have to touch these settings for 99.99% of your projects.

The configuration file basically contains following main elements −

Sr.No	Interceptor & Description
1	**struts-config** This is the root node of the configuration file.
2	**form-beans** This is where you map your ActionForm subclass to a name. You use this name as an alias for ActionForm throughout the rest of the strutsconfig.xml file, and even on your JSP pages.
3	**global forwards** This section maps a page on your webapp to a name. You can use this name to refer to the actual p This avoids hardcoding URLs on your web pages.
4	**action-mappings** This is where you declare form handlers and they are also known as action mappings.
5	**controller** This section configures Struts internals and rarely used in practical situations.
6	**plug-in** This section tells Struts where to find your properties files, which contain prompts and error messag