## Why Struts 2, Introduction To Struts Framework

Let us see the quick and brief introduction to struts 2 framework, struts is an open source framework given by Apache software foundation under one of its projects called Jakarta.  Struts is the frame work, used to develop web applications for java with mvc2 architecture.

Actually struts 1 was introduced in 2004 and this framework is failed to satisfy the needs of customers in today’s worlds,  finally Apache joined with OpenSymphony and created struts2.x.

struts 2 = webwork2 + struts 1

webwork2 is the framework from OpenSymphony,  java based mvc2.

You know what, struts 2 is not an extension of struts 1, its the combination of struts 1 and webwork2 some features taken from struts1 and some from webwork2 and finally released this struts2 frame work.

## Types of frame works

Frameworks are divided into 2 types,

* Invasive
* Non-Invasive

Invasive means, it will force the programmers to create their classes by extending or implementing from per-defined classes or interfaces provided by that frame work.  Non-Invasive means it wont forces the programmer to extend or implement its own classes or interfaces.

Struts is the type of Invasive frame work.

## Why struts so popular

* Struts supports extensive validations where other frame works doesn’t
* Having inbuilt support for I18N
* Struts 2 actions classes are spring friendly so we can easily integrate
* In build AJAX themes to make the applications more dynamic
* Finally good frame work for front end based applications, i will explain this point later hah

## Struts 2 based on MVC 2 architecture

MVC : Model View Control, am giving real time definitions, i don’t want to confuse you by saying lines of definitions :-)

**Model:** Will concentrates on business logic of an application, am going to take bean as model

**View:** will take cares the presentation, i mean visible part, we are going to use jsp as view for these struts 2 applications

**Controller**: is a component, which contains all primary logic’s required, like sessions and security and also it contains the entire flow of applications, we are going to use servlet as controller for these struts **2** applications.

Hope you are clear**,** don’t think much see the next tutorial am promising you that i will make you to understand this struts2 better than what you are expecting ;)

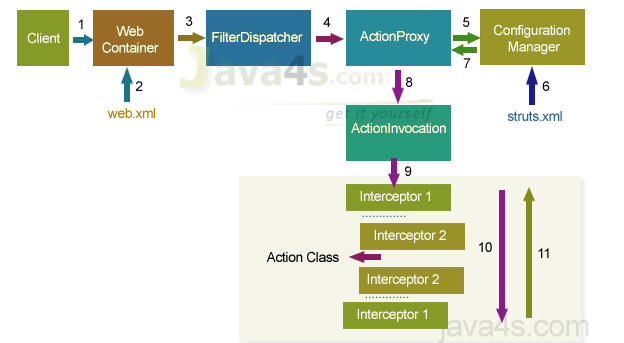
## Struts 1.x vs Struts 2.x Main Differences

* In struts 1.x front controller is ActionServlet
  + In 2.x front controller is FilterDispatcher
* In struts 1.x we have RequestProcessor class
  + In 2.x we have Interceptors instead RequestProcessor will see about this concept later just remember as of now
* In struts 1.x we have multiple tag libraries like, html, logic, bean..etc
  + In 2.x we do not have multiple libraries, instead we have single library which includes all tags
* In struts 1.x the configuration file name can be [any name].xml and we used to place in web-inf folder
  + In 2.x the configuration file must be struts.xml only and this must be in classes folder
* In struts 1.x we have form beans and Action classes separately
  + In 2.x form bean, Action classes are combinedly given as Action class only, of course we can take separately if we want ;)
* In struts 1.x properties file must be configured in struts-config.xml
  + But in 2.x we need to configure our resource bundle(s) in struts.properties file
* In struts 1.x we have programmatic and declarative validations only
  + In 2.x we have annotations support too along with programmatic and declarative validations

## Functional Differences

* In struts 1.x declarative validations are done by using validation frame work
  + In 2.x, declarative validations are done by using xwork2 frame work by webwork the reason being, its support valuations through Annotations
* In struts 1.x an Action class is a single ton class, so Action class object is not a thread safe, as a programmer we need to make it as thread safe by applying synchronization
  + In 2.x an Action class object will be created for each request, so it is by default thread safe, so we no need to take care about safety issues here
* In struts 1.x we have only **jsp** as a view technology
  + In 2.x we have support of multiple view technologies like velocity, Freemarker, jasper reports, jsp bla bla
* In struts 1.x Action class is having servlet dependency, because in execute() method accepts req, res parameter right ! so.
  + In 2.x Action class doesn’t have any servlet dependency, because its execute() method doesn’t accepts any parameters, however we can access all servlet objects with dependency injection :-)

## Struts Execution Flow Diagram, How Struts Works



## Execution flow of struts

* When a client request is given, a web container will receive request
* Web container loads web.xml and verifies whether the url-patterns are verified or not, if matches web-container transfer the request to FilterDispatcher
* FilterDispatcher hand overs the request to ActionProxy, it is a proxy class which is responsible to apply before and after services to original business logic
* ActionProxy contacts ConfiguraionManager class, to know the suitable Action for the request and the needed services for the request
* ConfigurationManager class loads structs.xml and provides the required information back to ActionProxy
* ActionPorxy delegates the request along with its information to ActionInvocation
* ActionInvocation executes the interceptors added to an Action from 1 – N, after that it will call the business logic implemented from N – 1 in reverse order
* ActionInvocation receives finally result produced by an action aclass
* ActionProxy transfers the result back to FilterDispatcher
* FilterDispatcher selects an appropriate view, basing on the result
* Finally FilterDispatcher uses RequestDispatchers forwarding mechanism and forward a view as a response back to the client

## Aware Interfaces of struts 2

In struts2 we don’t have any http specific objects by default just like in servlets.  If at all we want any http related objects in our Action class then we need to implement our Action class from Aware Interfaces provided by struts framework.

We can say here frame work using one form of dependancy injection of type interface for injecting required objects into an Action class of struts2.

Struts 2 provided total of 5 Aware Interfaces

* ApplicationAware Interface
* SessionAware Interface
* ServletRequestAware Interface
* ServletResponseAware Interface
* ParameterAware Interface

Every Aware interface provides a setter method, so we must override that setter method while implementing the particular Aware interface, at run time struts 2 controller will automatically calls that setter method and injects the required object into that Action class