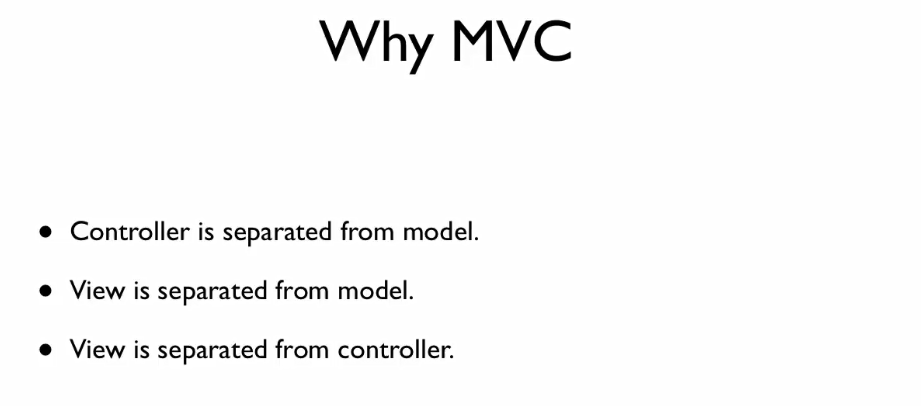
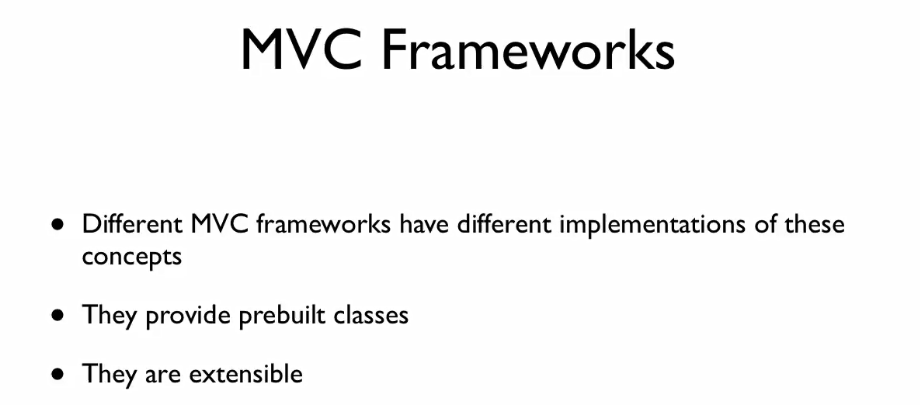
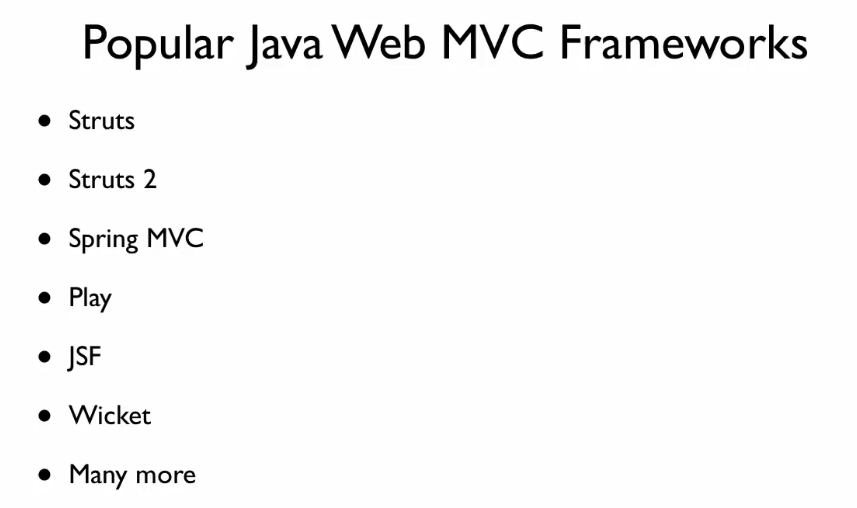
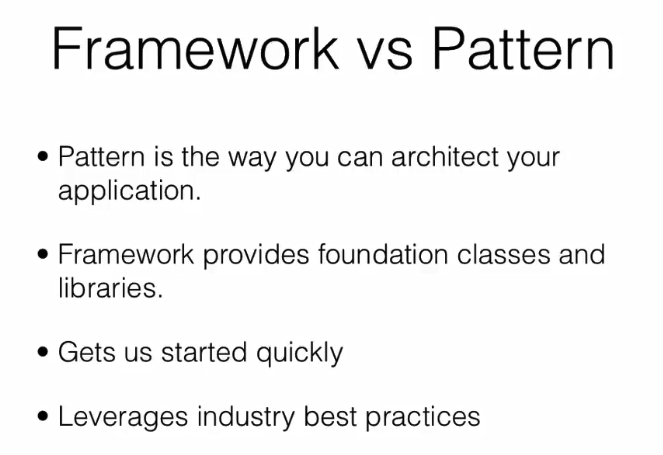
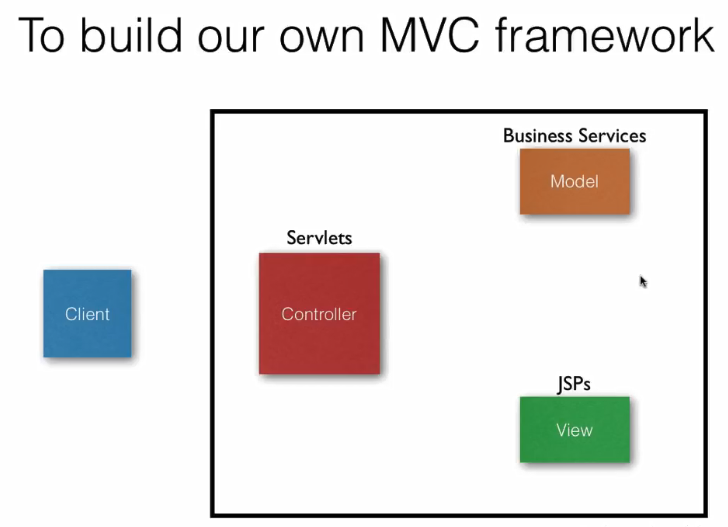
**Struts 2**

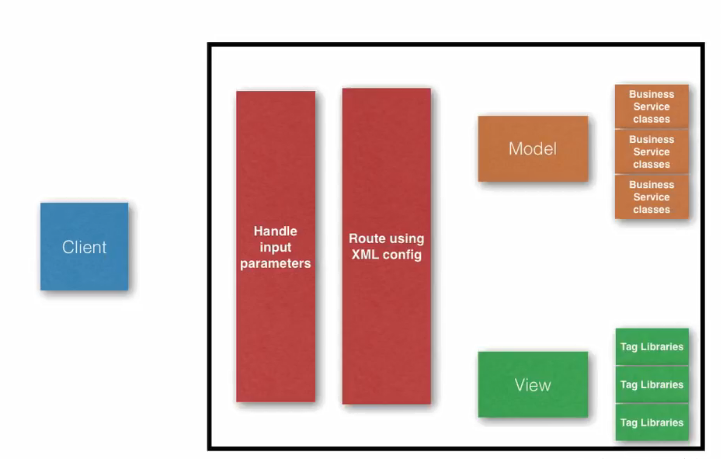
First of all shall we identify what is the MVC design architecture?

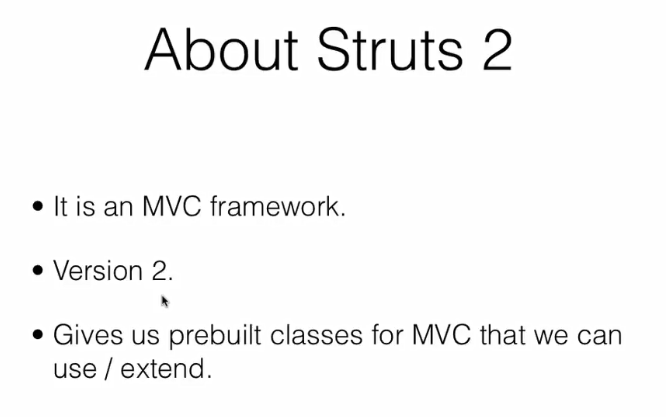


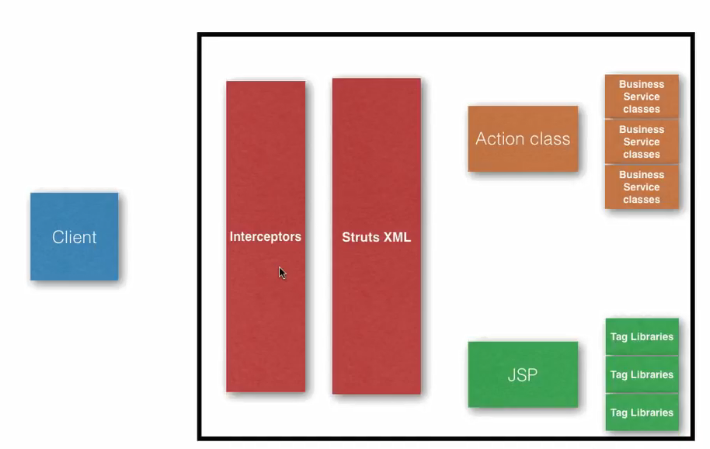




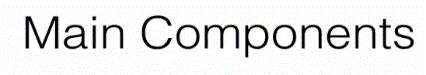


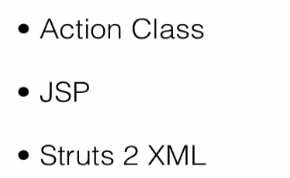


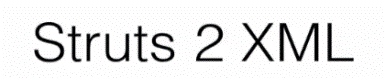


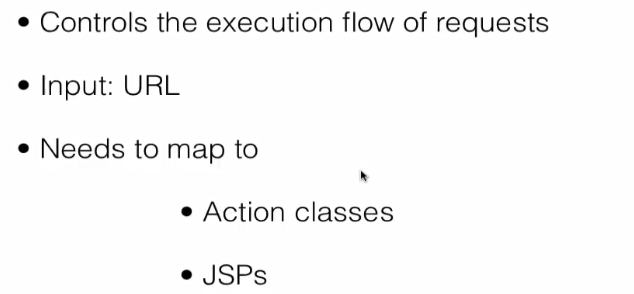


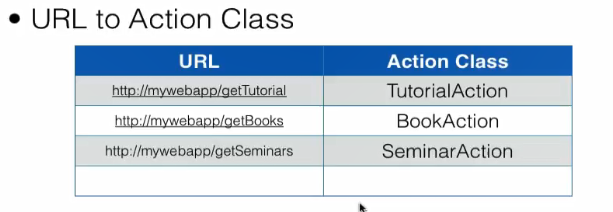


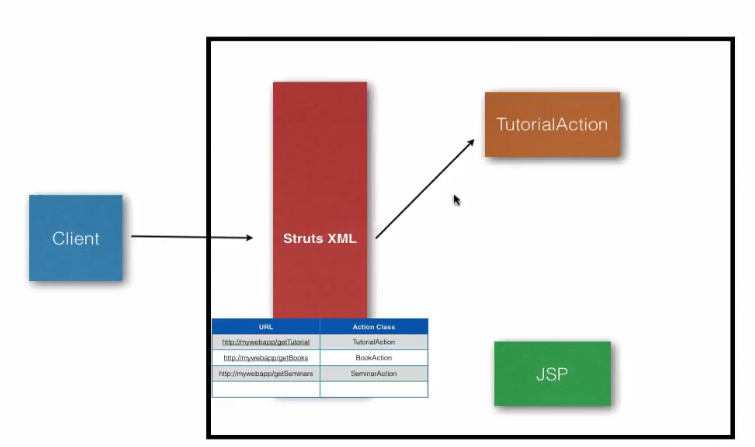




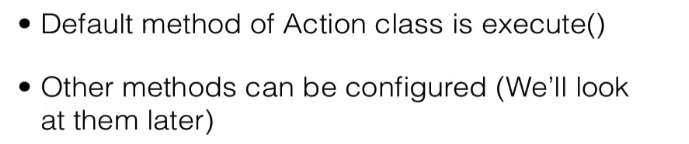


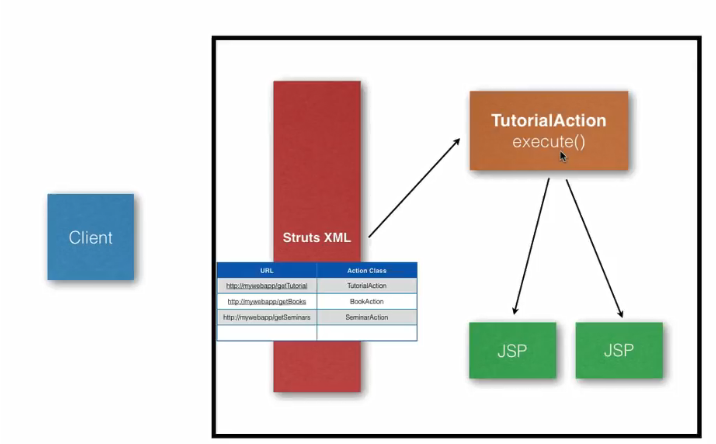


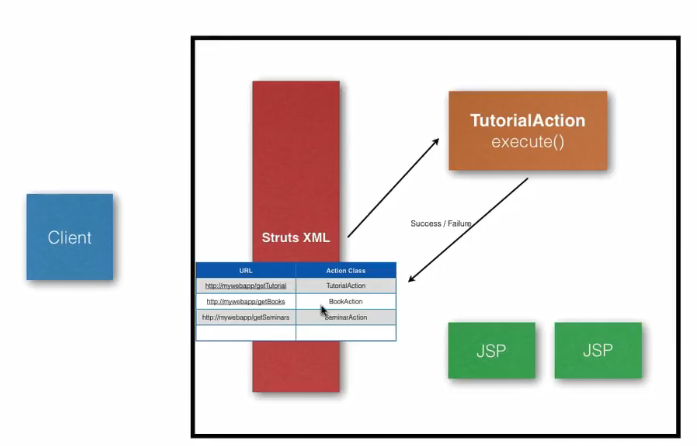


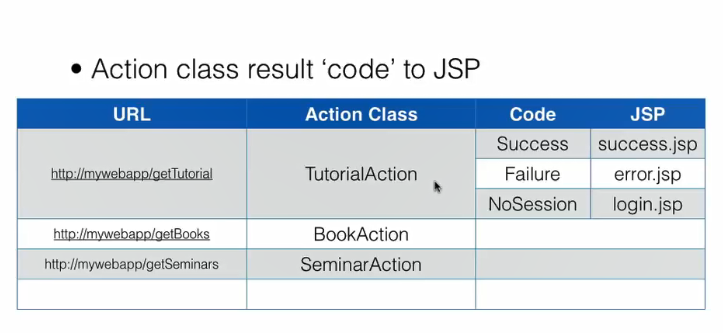


**Which method will be executed in action class?**



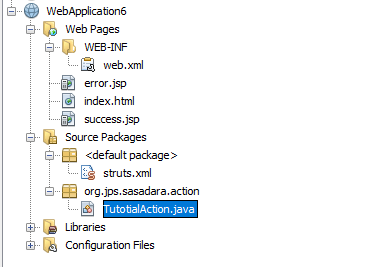


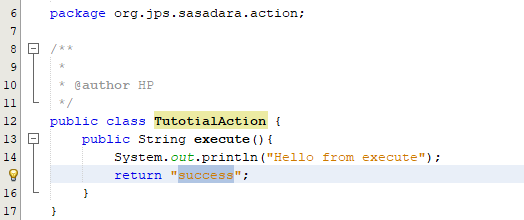
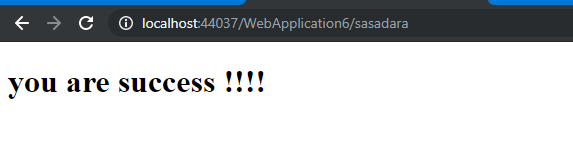




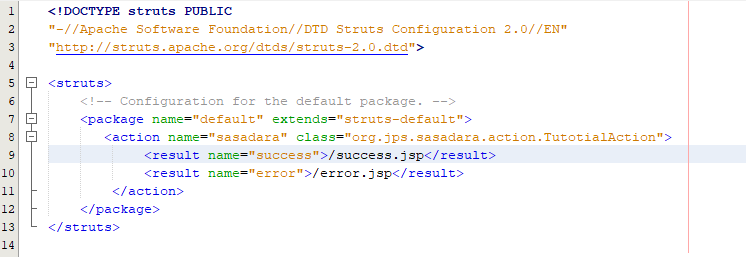
Example -:

**File structure -:**





4



1

7

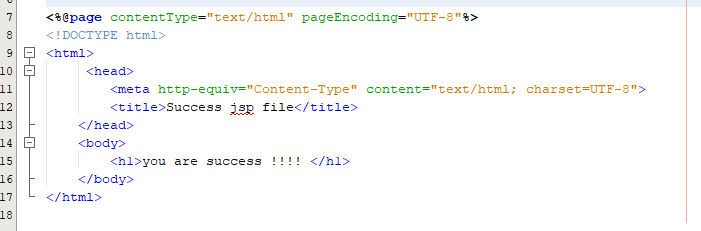
6

5

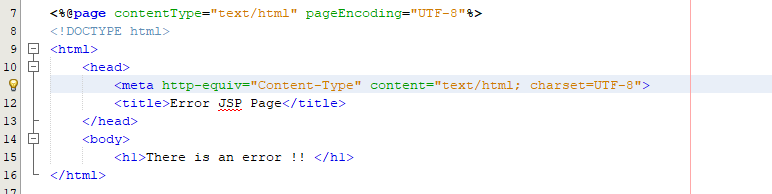
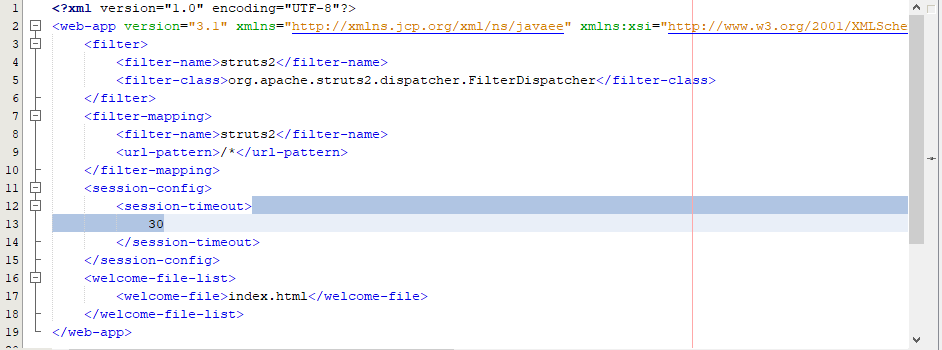
3

2





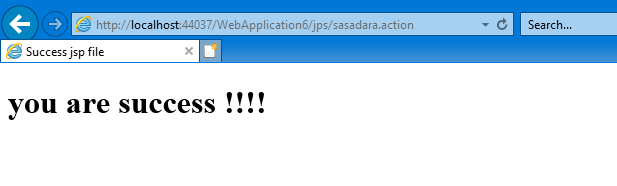
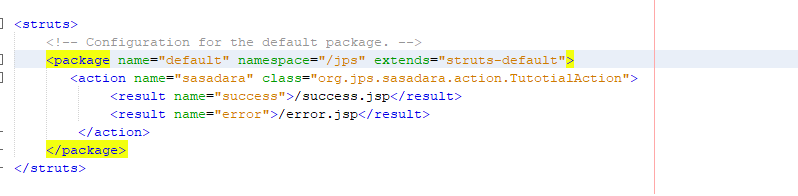


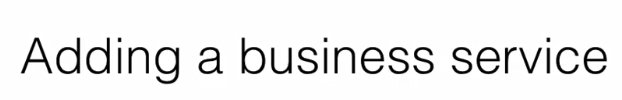


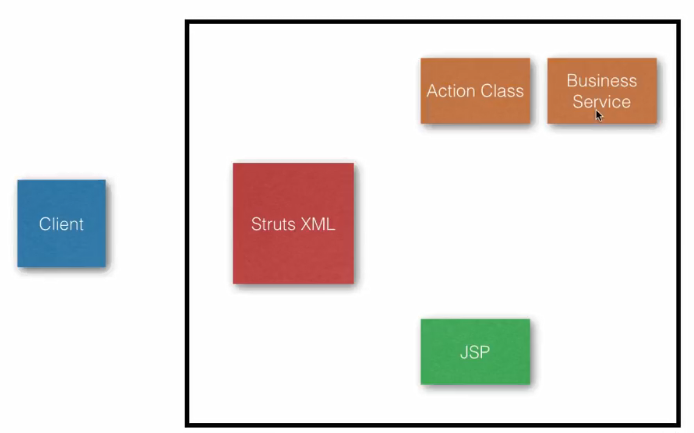
Filter are used to say look at the struts 2 routers

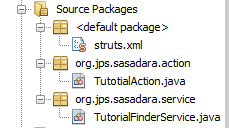
**How to use name space in struts 2**

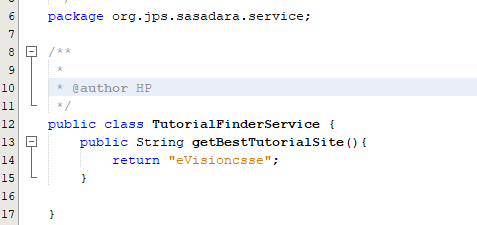
Example : -

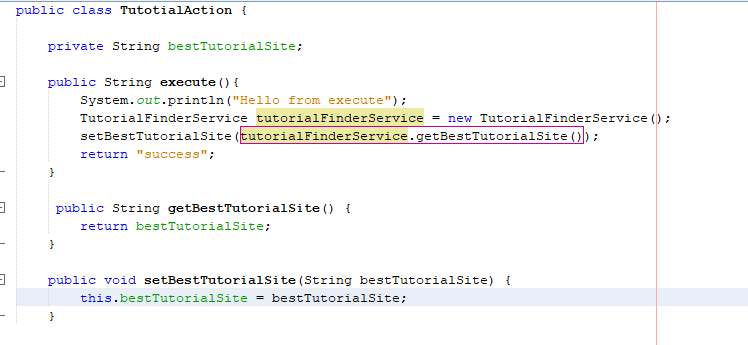






First shall we implement the business class within new package?

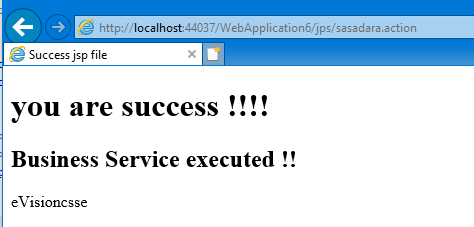


Next Service class should be called within the action class and let’s take return value in to new variable because it is important to keep that value in separate variable to access it inside the jsp page

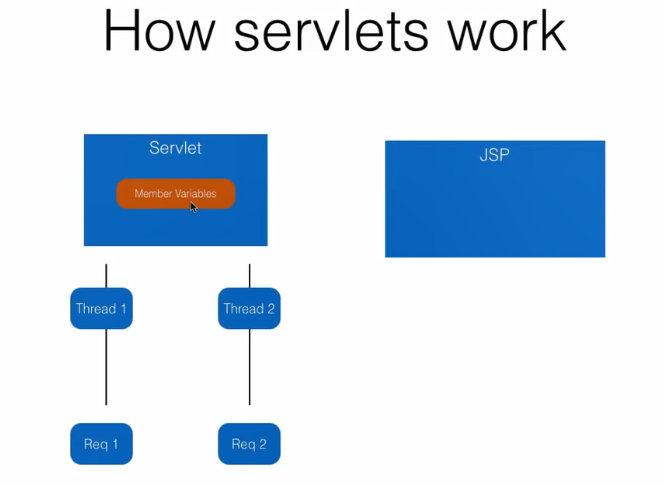
Then we can access that value inside the jsp file no need of using session

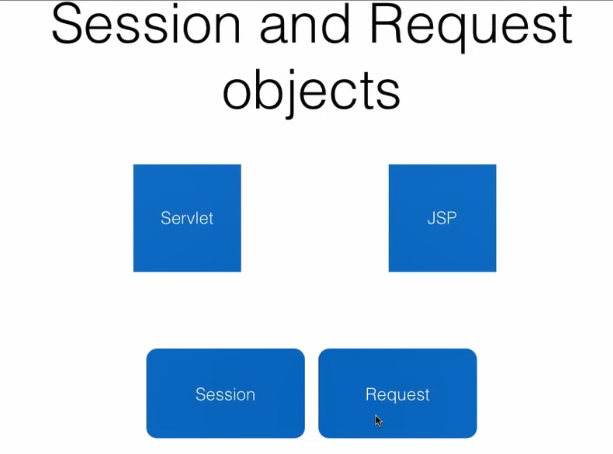


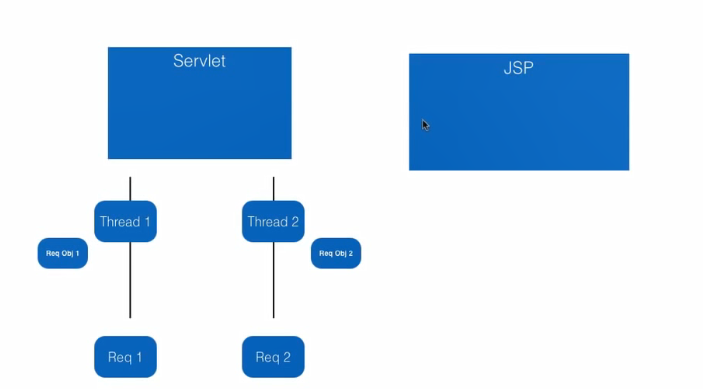
Lib file reference variable



**? ? How to JSP accesses the action class’s member variable .. Now let’s consider about the magic behind it …**

First let’s look at java EE servlet then we consider about Strats 2 action class

When we consider about servlet, if two request are coming to the same servlet, two instances will not be created to access those requests so that two thread will be created to access that servlet. So in this scenario Member variables are not thread safe. If we use servlet member variable to keep servlet data and access data inside the JSP, A critical region can be created by member variable. So in Servlet usages, we use two methods to keep and pass data to others.

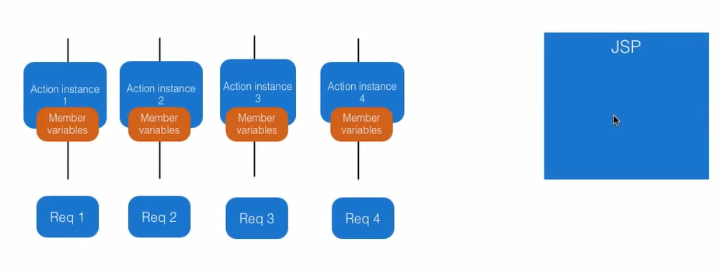
We can add data to request object and can pass them to others.

I have explained about “Request dispature ” in my Servlet Jsp A to Z tutorial you can refer it for more details…………

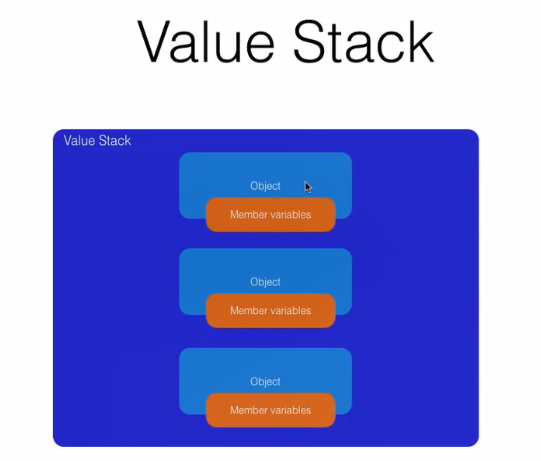
**In strats 2**

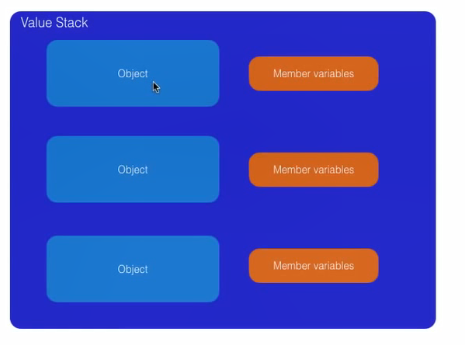


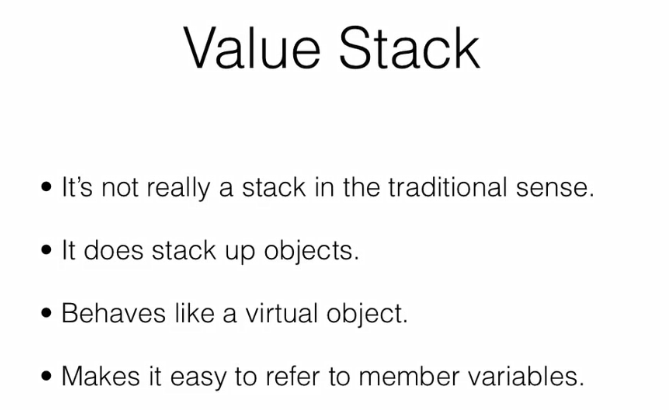


So we can keep and access data in the member variables since those are thread safe each request has separate member variable since separate action objects have been created for each requests in strats 2.

So every requests have separate “**Value Stack**” every created action instance of particular request has been keep inside that value stack.

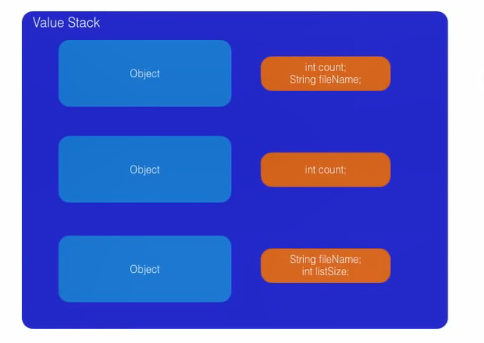


In the value stack all the member variable of each instance will act as member variable of Value stack not as member variable of instance (object).



So we can access member variable without considering the objects. But there can be issue even if there are two variable which have same member variable name.

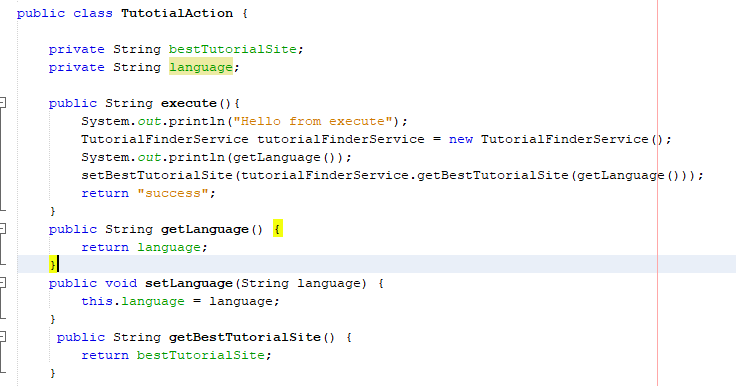
Eg-:

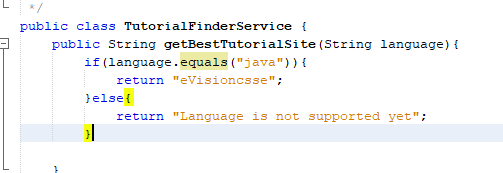
 In the scenario it will always access first member variable in this “Value stack”. So if we want to access second member variable, we can access it using <objectName>.<member\_variable>. We will talk about variable calling method in detail later.

**How to Access Input parameters**

1. Create member variable inside the particular action class (both should be same name).
2. Generate the getters and setters.
3. As previous those will be stored is value stack automatically.
4. Then we those variables can be access inside the JSP file.

Ex-:





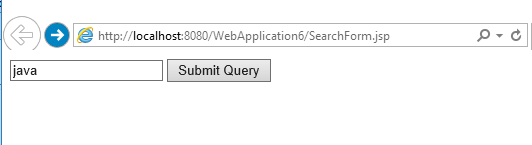


**Post Request to action**

Same as get parameter, post parameter can be accessed by member variable of action class and keep them inside the value stack.

Eg-; Using HTML

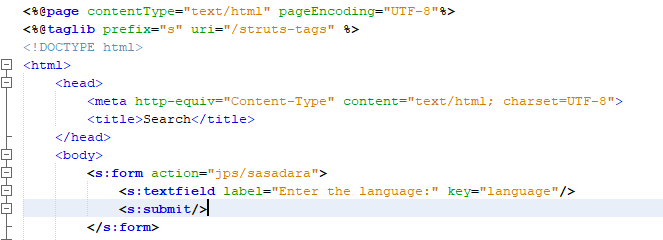




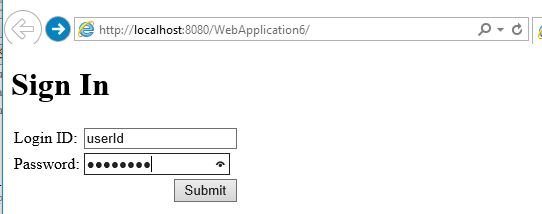


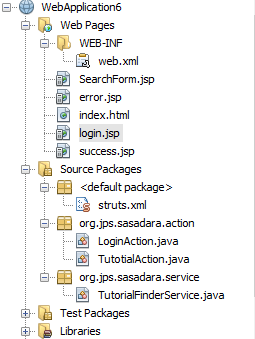
Eg-: Using Taglib of Strats 2

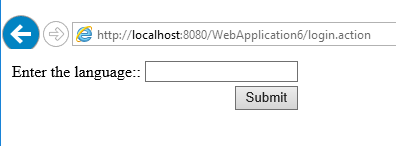




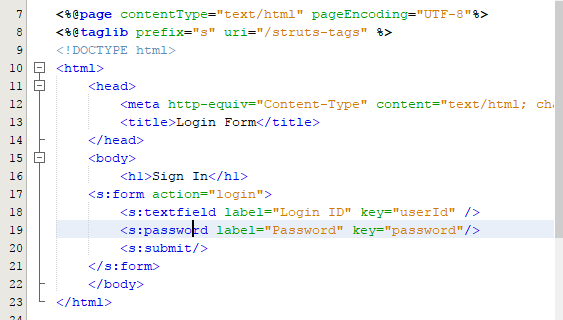
**Login page Example:-**





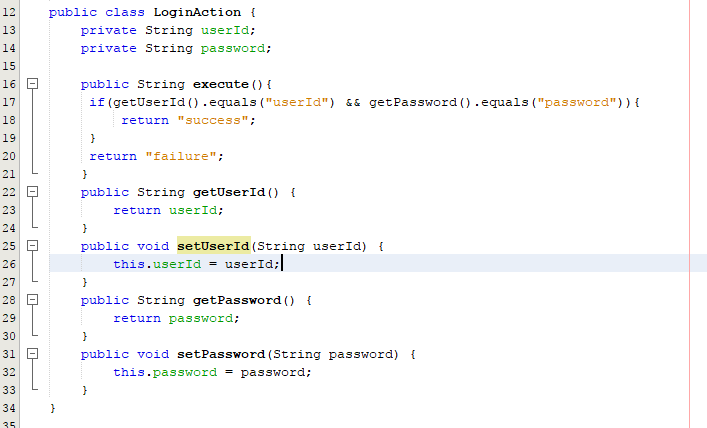




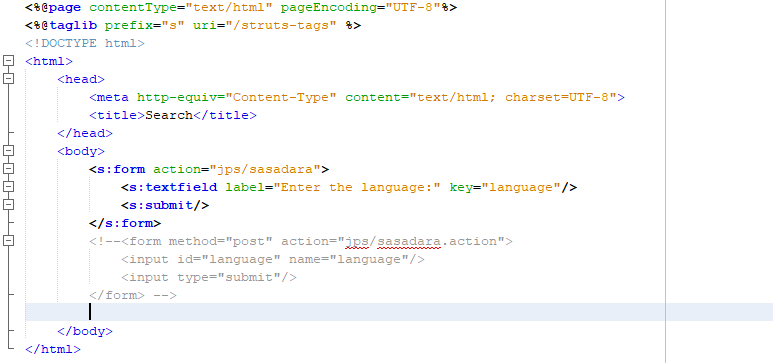




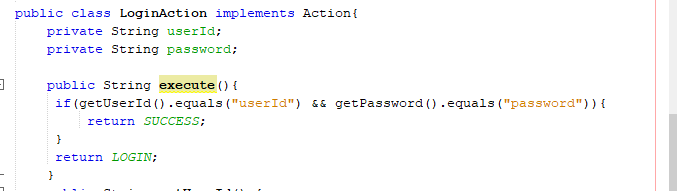


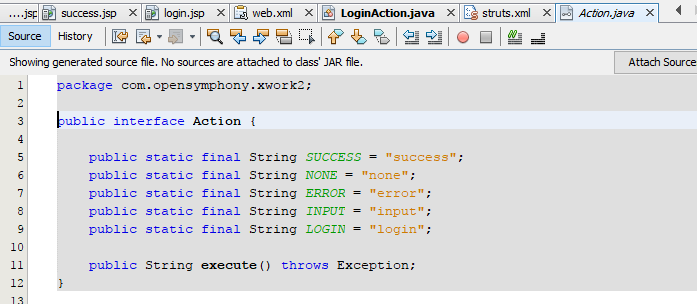
 



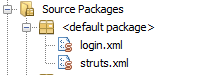


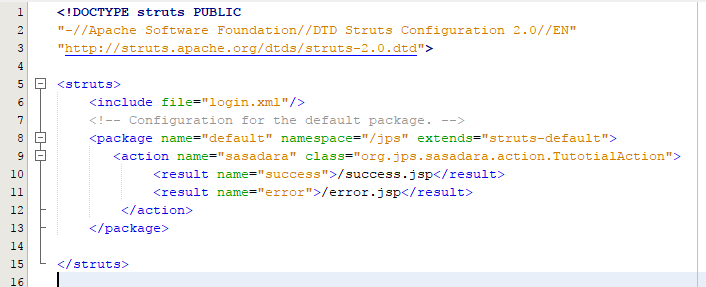
**Shall we add three best practice for this login example**

1. **Use variables of “Action” interface**

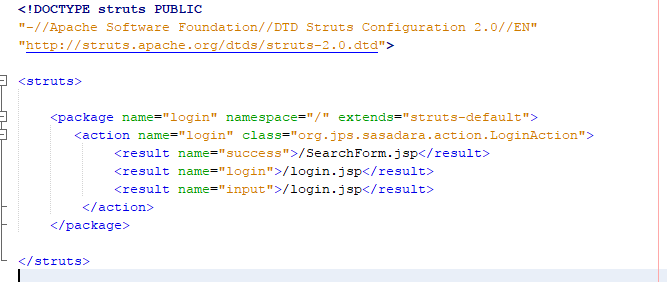


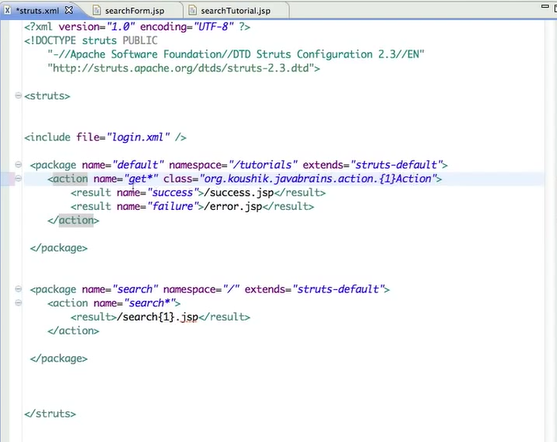
1. **Divide struts.xml file several.**

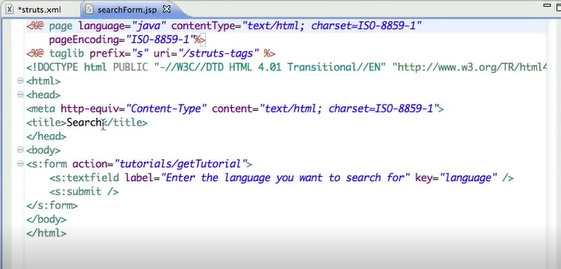


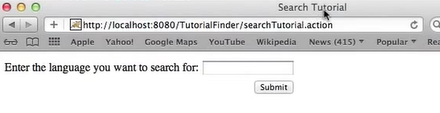






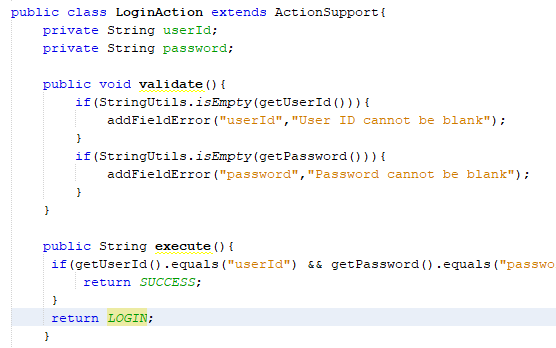
**Action Wildcards**



**ActionSupport class**

In the “ActionSupport class” there is a method called “Validate(){}” by using this method we can implement the validations. This “Validate(){}” method will be executed before run the “execute(){}” method of particular action class

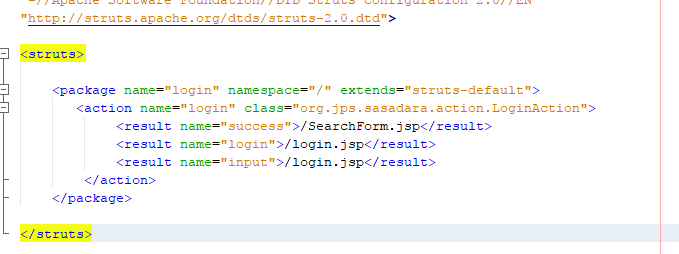


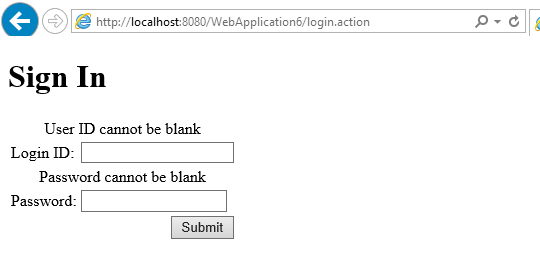


If there any value was added to the “addFieldError()” method 🡪

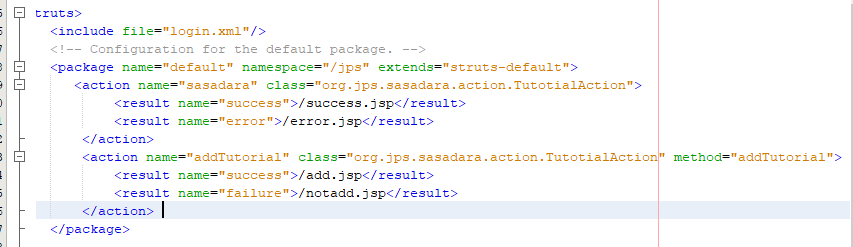
The “execute()” method will not be executed and internally it will return a Sting called “input” it will help us to configure relevant JSP file inside the struts.xml file . so it will show the error messages that we have defined before the particular text boxes.

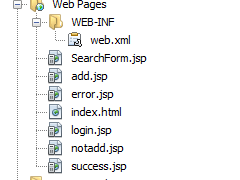
In this our example 🡪



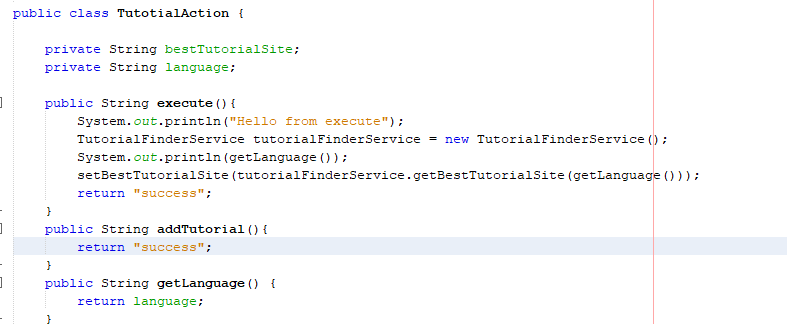


**Configuring Methods in Action Mapping**

We can configure an our own method to execute without using the default “execute()” method

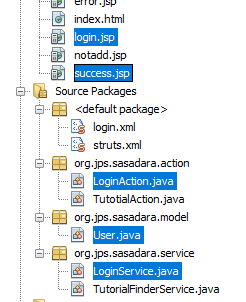








**How to use model object**



**related files have been highlighted**



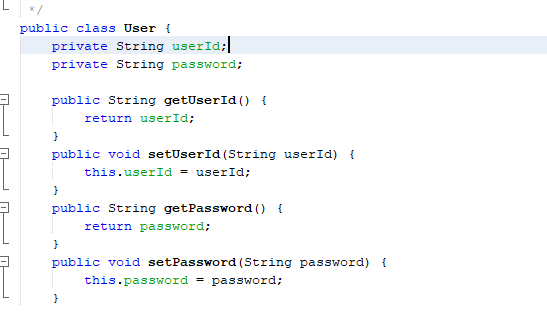
**Value Stack**



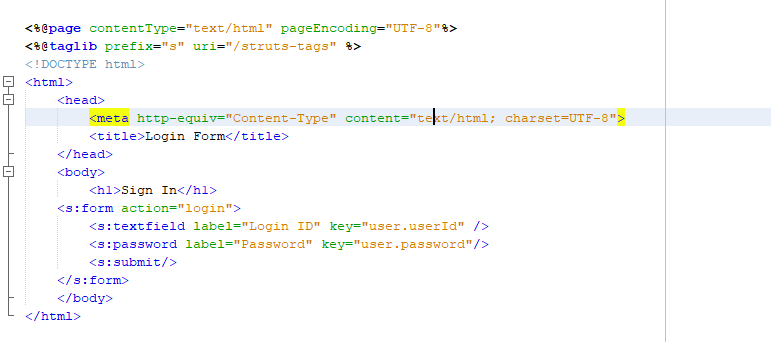


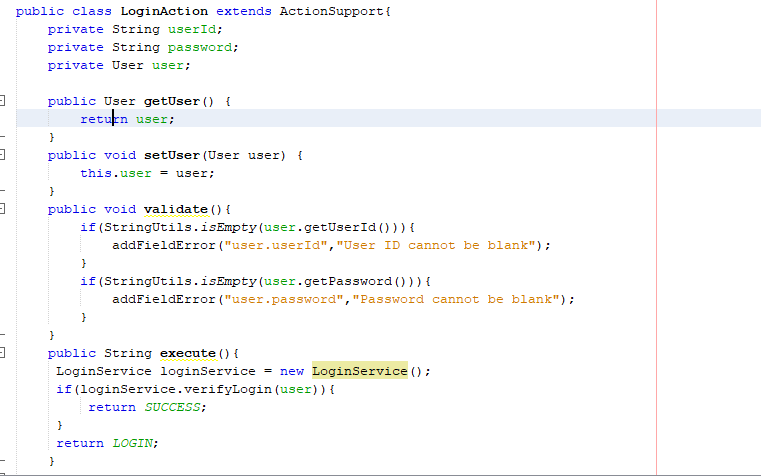


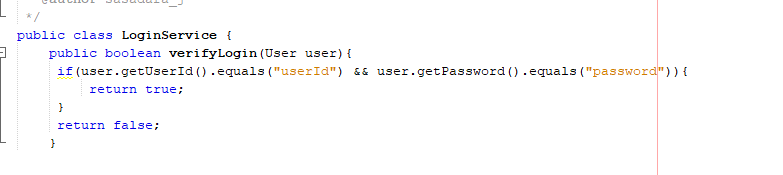


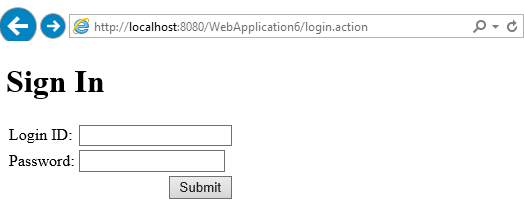












**Second method is model driven**

