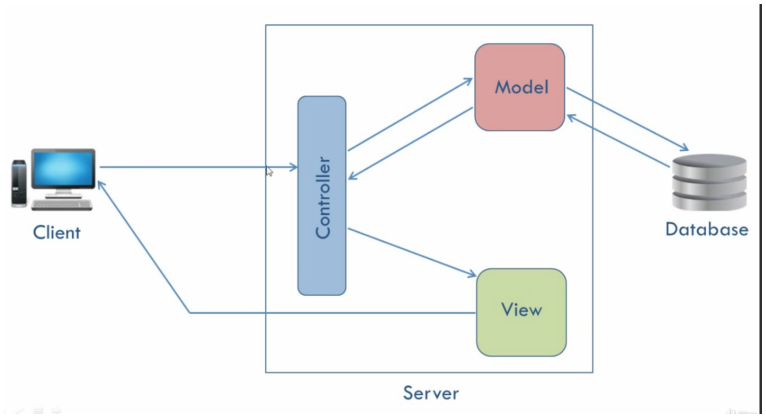
What is Struts 2?

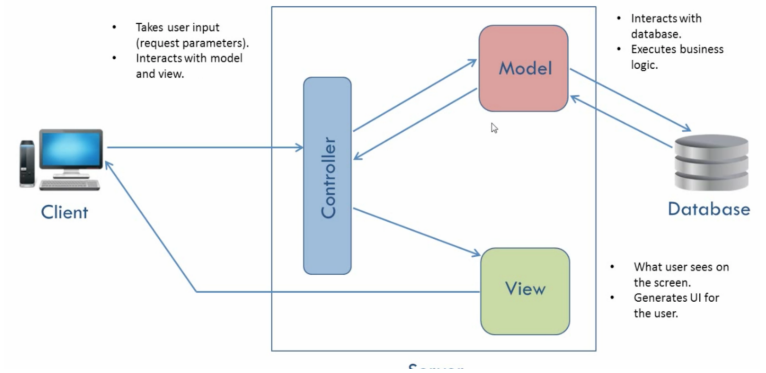
- Apache Struts 2 is a web application framework used for developing Java EE web applications.
- It implements MVC pattern.
- It is one of the popular Java web framework.

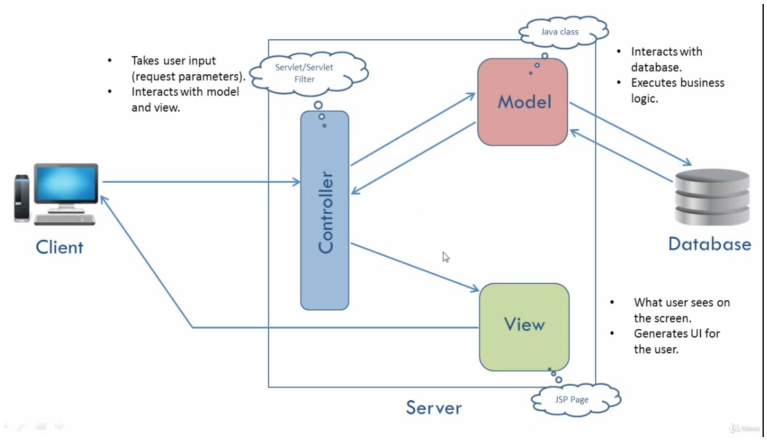
Overview

- What is a Framework?
- What is MVC?
- Struts 2 Architecture
- Registration Application
- ValueStack and OGNL
- Interceptors
- Struts 2 Tags
- Internationalization
- Integration with Database

- MVC stands for Model View Controller.
- It is an architectural pattern.
- MVC divides a software application in 3 parts:- model, view and controller.

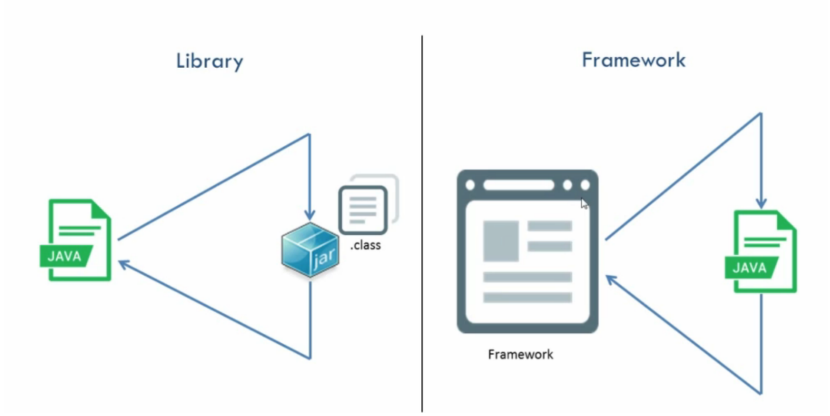






What is the framework and its benefits?

- A framework is a collection of some foundation classes and libraries which makes application development easier.



Library

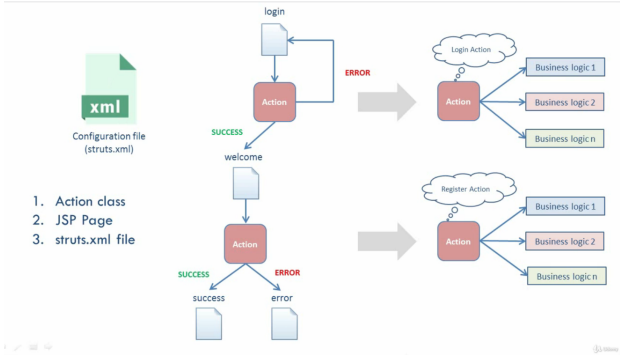
- Our application code makes a call to the library.
- Our application code is in control.
- Library is reusable code.

Framework

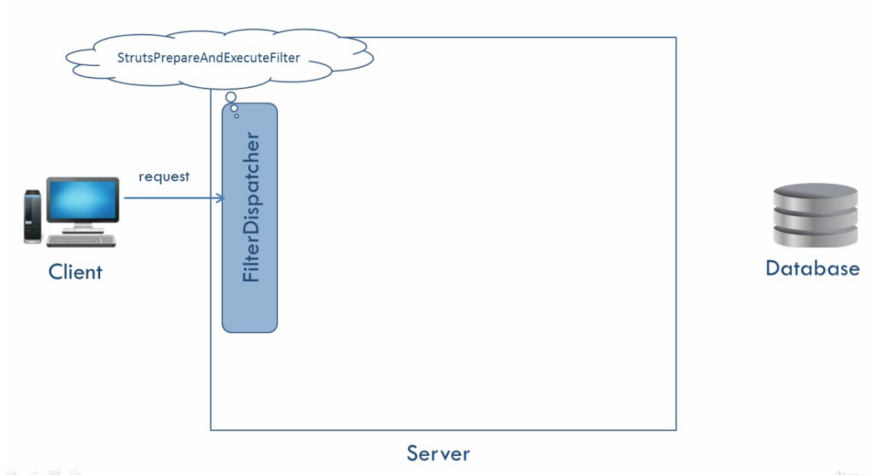
- The framework makes a call to our application code.
- The framework is in control.
- Framework is reusable architecture.

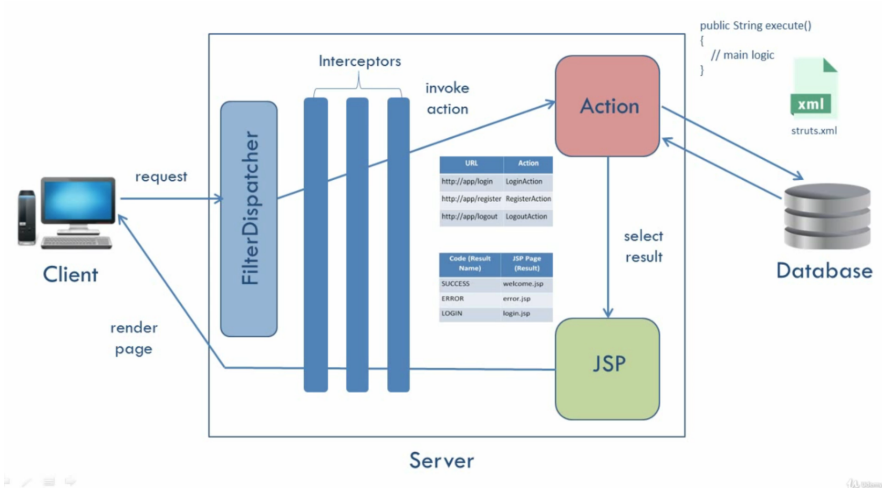
Benefits of a Framework

- Framework makes us write our application by following a pattern.
- We don't have to worry about the design of our application.
- Makes our application maintainable in future.



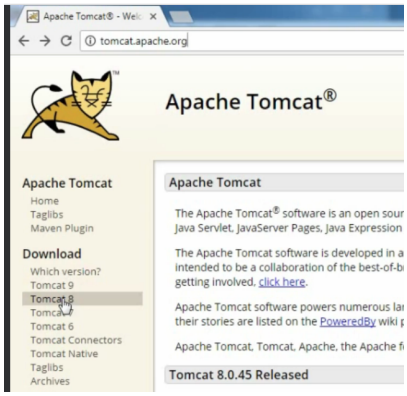
Each action will have an execute method.

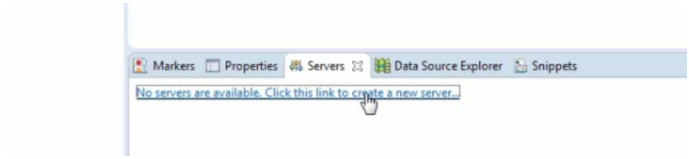


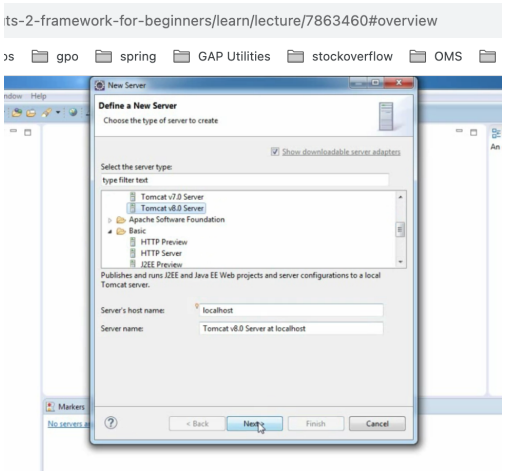


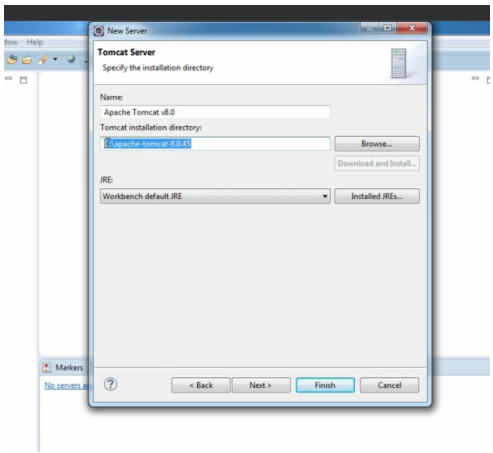
Every action is an execution method.

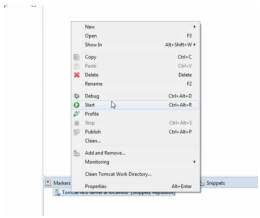
Setup Tomcat Server

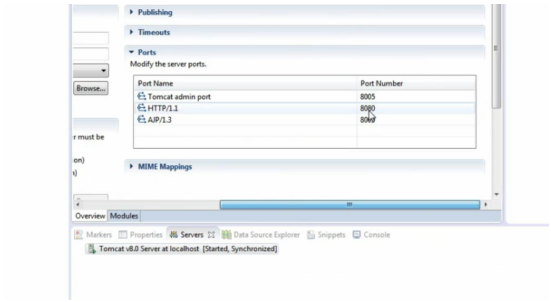



Markers
Properties
Servers
Data Source Explorer
Snippets
No servers are available. Click this link to create a new server...


its-2-framework-for-beginners/learn/lecture/7863460#overview
gpo
spring
GAP Utilities
stockoverflow
OMS
New Server
Define a New Server
Choose the type of server to create
Show downloadable server adapters
Select the server type:
type filter text
Tomcat v7.0 Server
Tomcat v8.0 Server
Apache Software Foundation
Basic
HTTP Preview
HTTP Server
J2EE Preview
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.
Server's host name:
localhost
Server name:
Tomcat v8.0 Server at localhost
Markers
< Back
Next >
Finish
Cancel

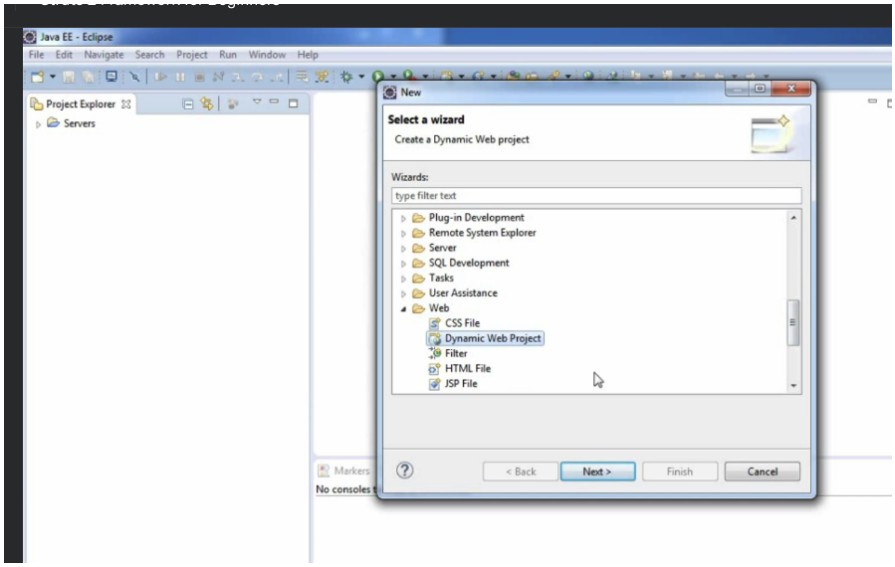

New Server
Tomcat Server
Specify the installation directory
Name:
Apache Tomcat v8.0
Tomcat installation directory:
Browse...
Download and Install...
JRE:
Workbench default JRE
Installed JREs...
Markers
< Back
Next >
Finish
Cancel

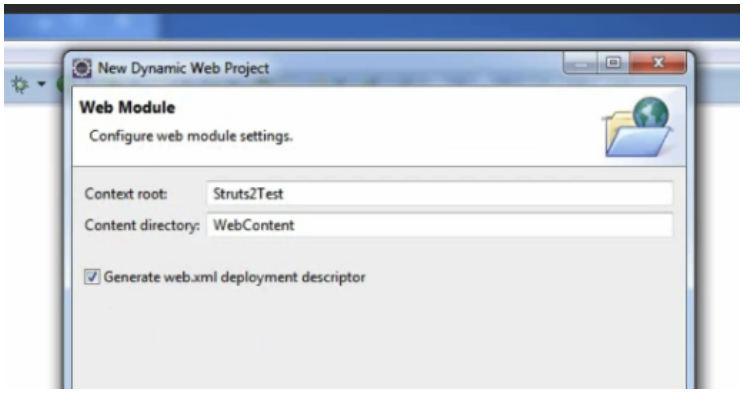


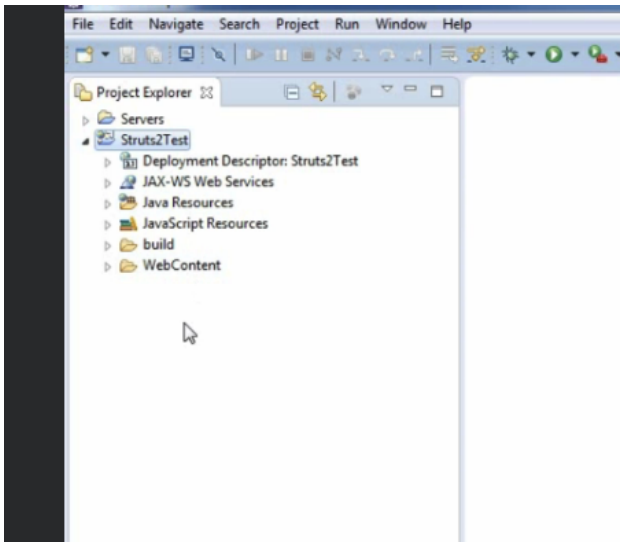


Double click if you want to change the port number

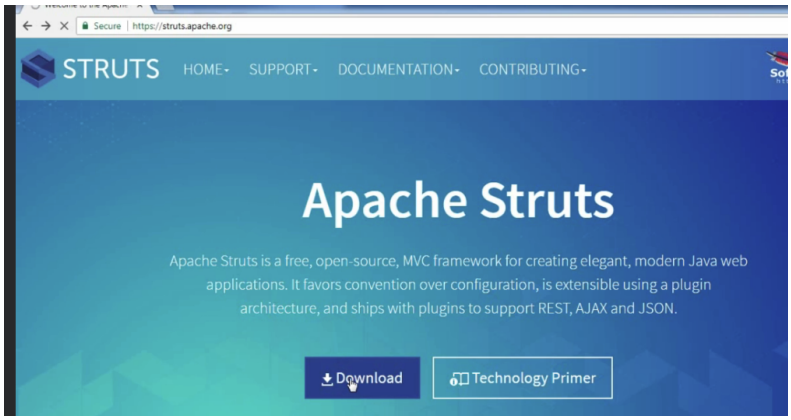
Setup Struts 2 work environment

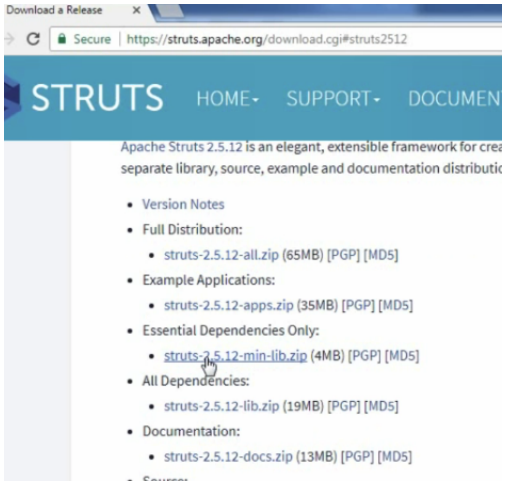




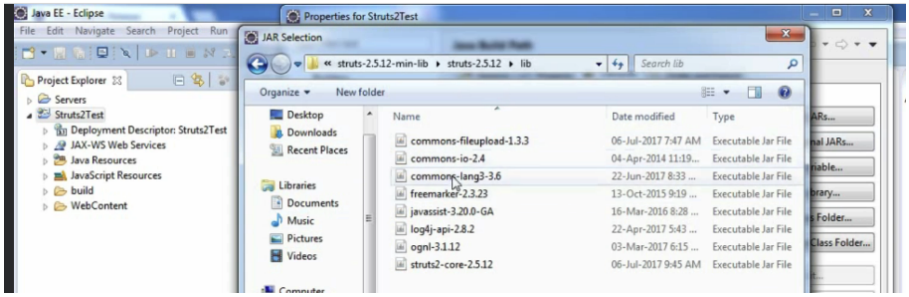


Download struts jar

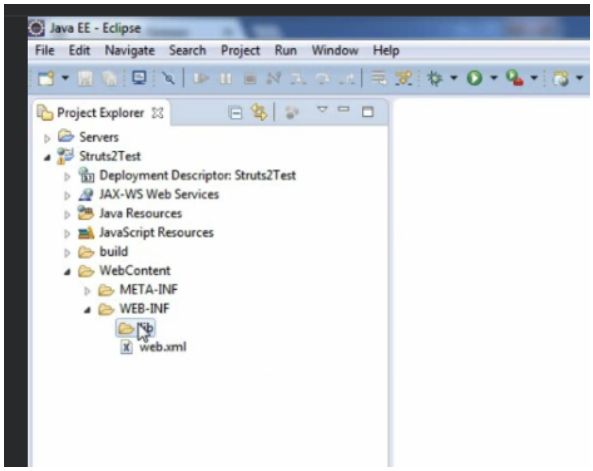


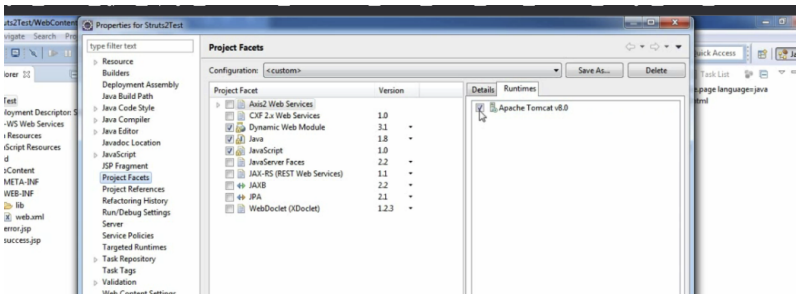


Download and add to to the proj



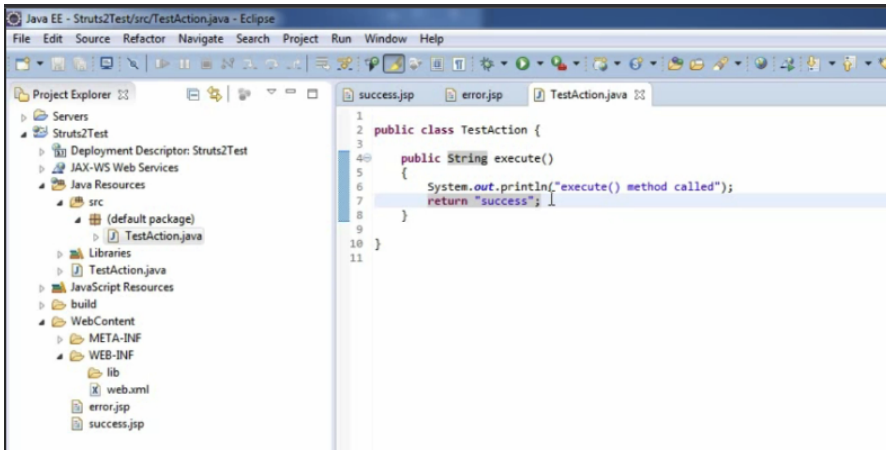
Add same thing to below lib folder as well.



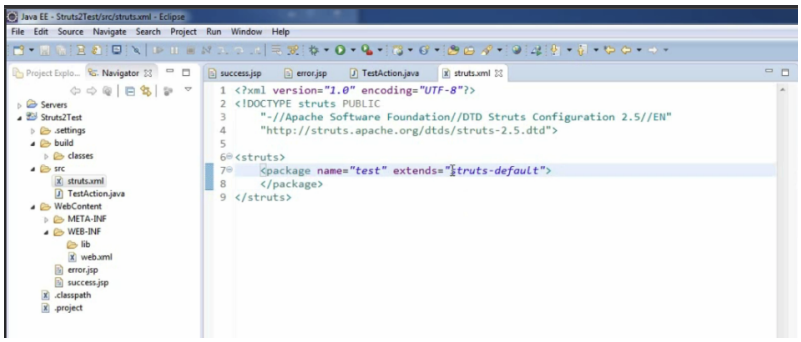


Ensure that check box enabled from project folder

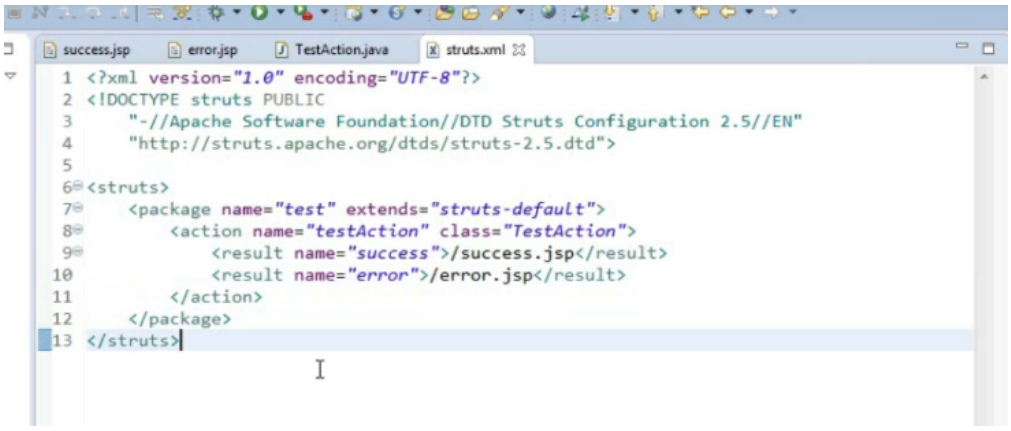
Create success and error jsp. And create an action class. Each action class contains an execute method.
Each execution method contains business logic as shown below



Now we need struts xml in the src folder as shown below. With line 2, 3, 4 this xml will be treated as struts.xml.

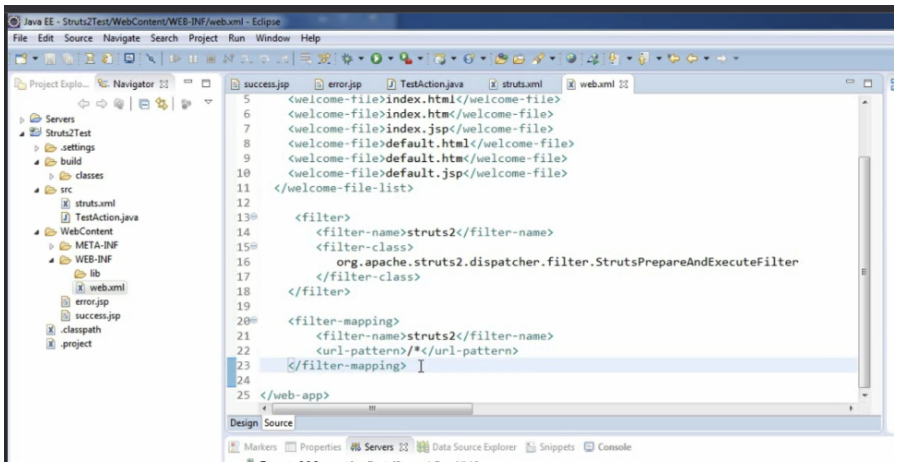


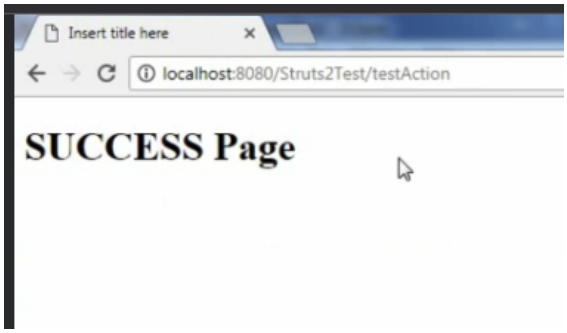
Package groups actions and result types into logical unit as shown below.



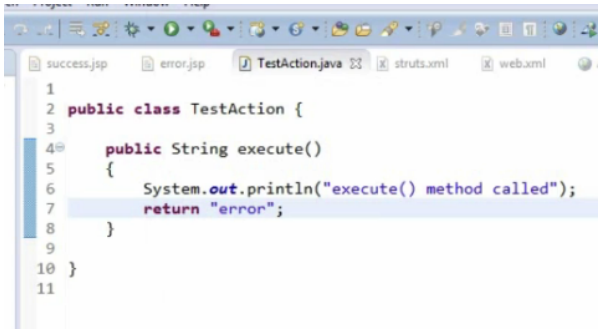
<action name=”testAction” is path url, class=TestAction which java class execute method should execute. Here TestAction.java class execute method will be executed.

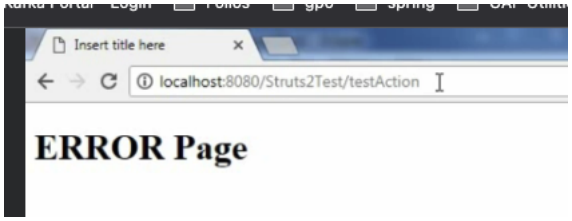
Each action will have <result name = what execute method returns that should match here to /success.jsp
So far we did model and view. Now the controller filter. So everything will come first to filter and redirect to based on url pattern





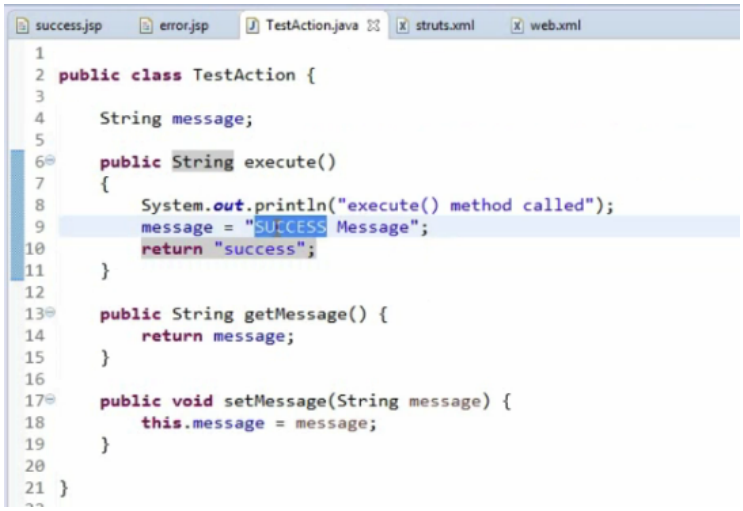
Change return type to error



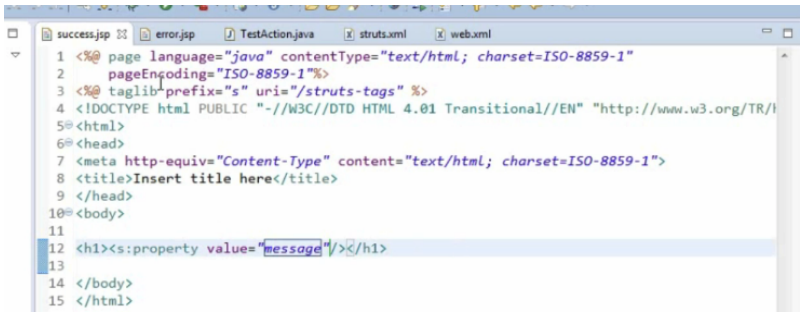


10. Passing data from Action to JSP

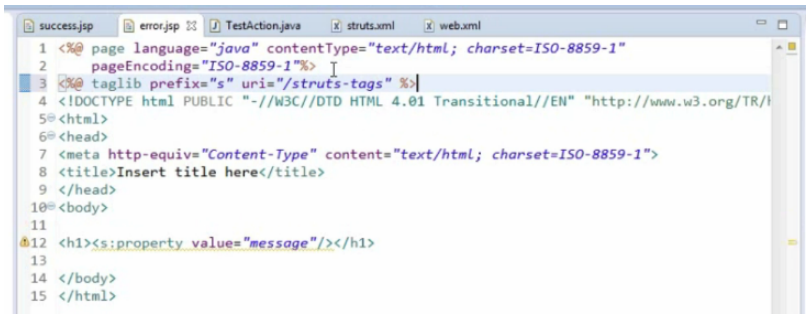
In our action class introduce a variable message and getter and setter. And logic in execute method as shown below.



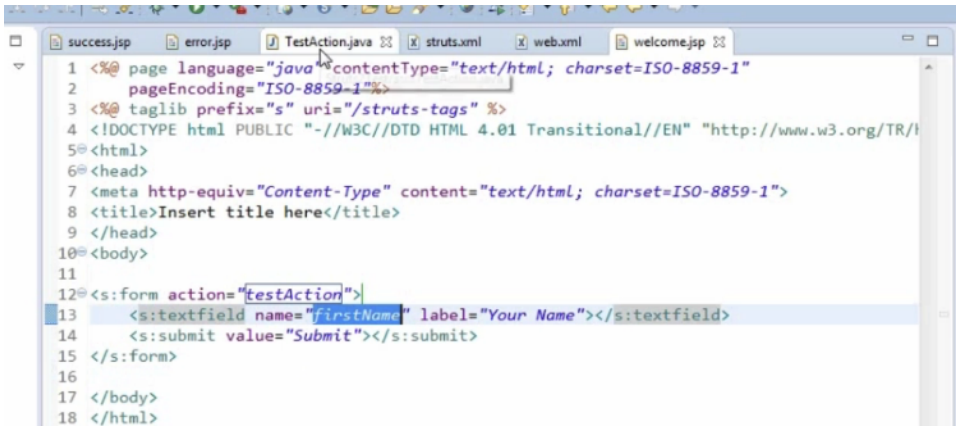
Add struts-tags to the jsp page. And introduce a property tag at line 12. We are given a value as “message” which is used in our action class that will return here.



Similarly for error.jsp

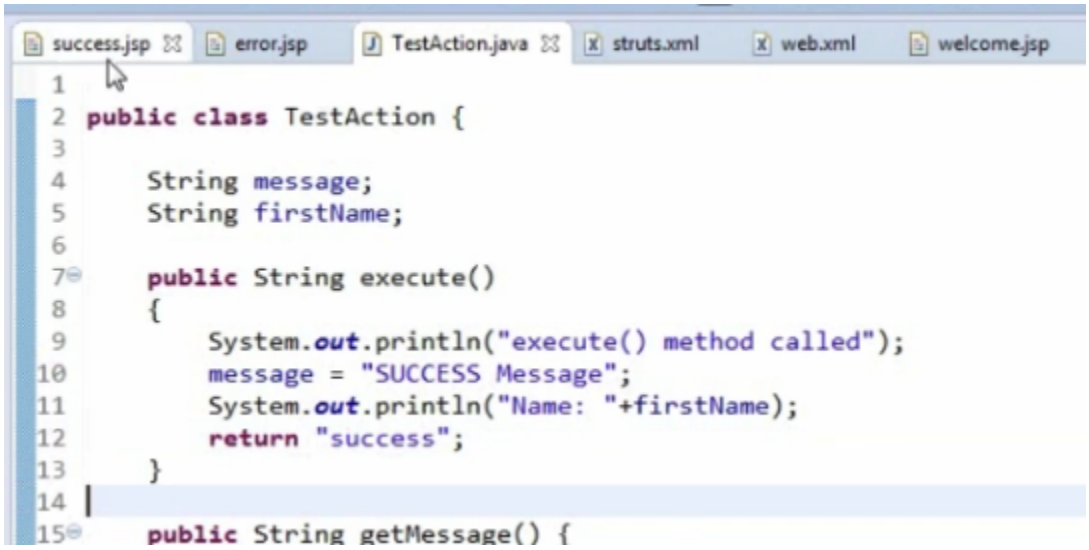


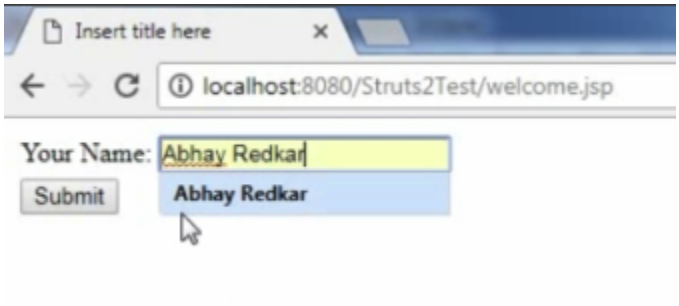
Create a welcome page which contains a text field and submit button.



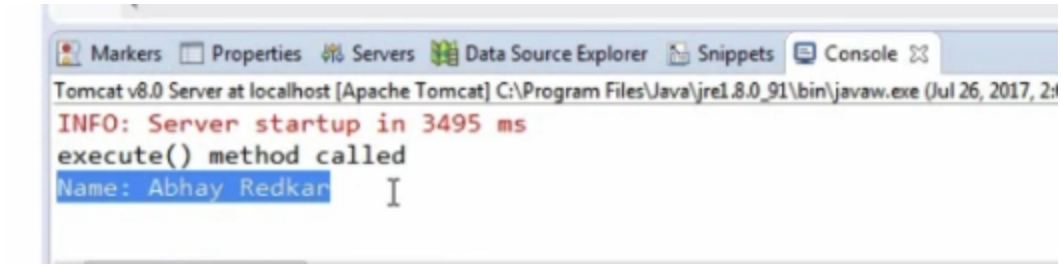
On submit it will call testAction.java class.

Create same name variable in action class with setter and getters

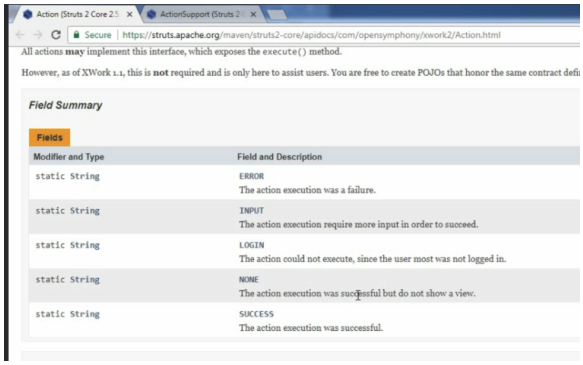




Those two print statements printed

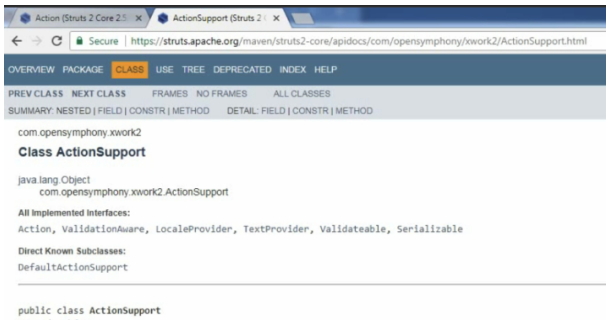


Action interface and its fields. We can use them in return statements.



Here we are calling the SUCCESS field.

We also have an ActionSupport class to extend our action class that will give access to more useful methods. Can be use used a interceptor

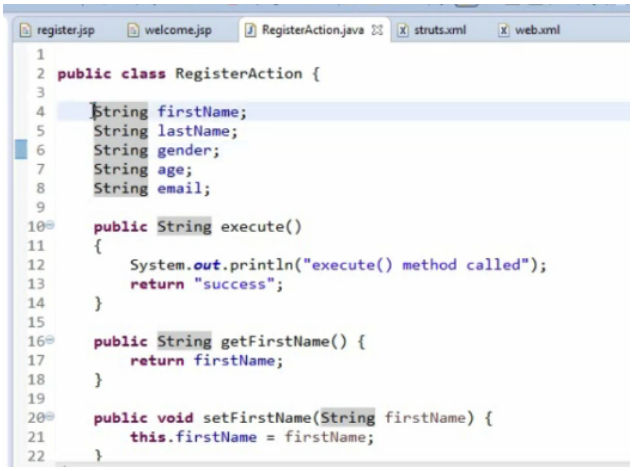


register.jsp | welcome.jsp | RegisterAction.java | struts.xml | web.xml

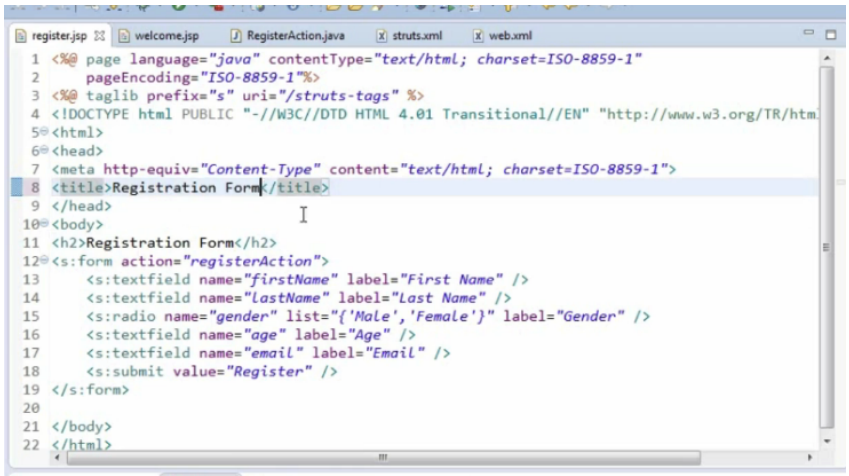
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">

<struts>
    <package name="register" extends="struts-default">
        <action name="registerAction" class="RegisterAction">
            <result name="success">/welcome.jsp</result>
        </action>
    </package>
</struts>
```

Struts package update registration action should return welcome page on success.

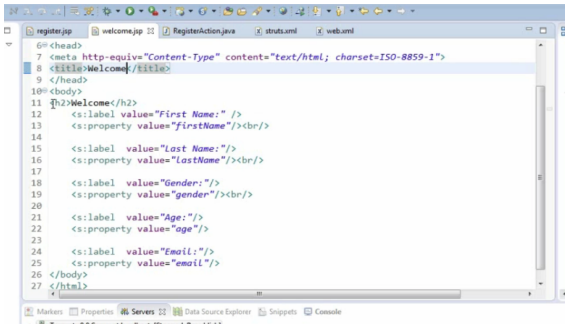
Different UI components in RegistrationAction java file

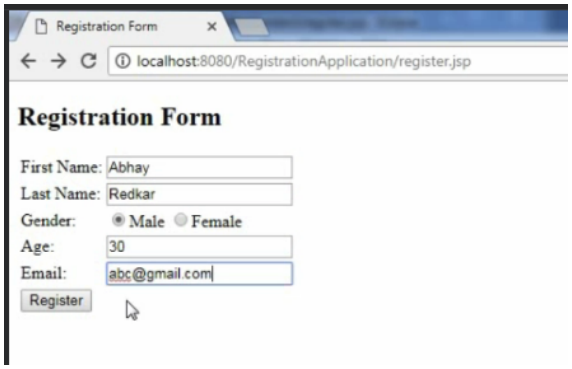


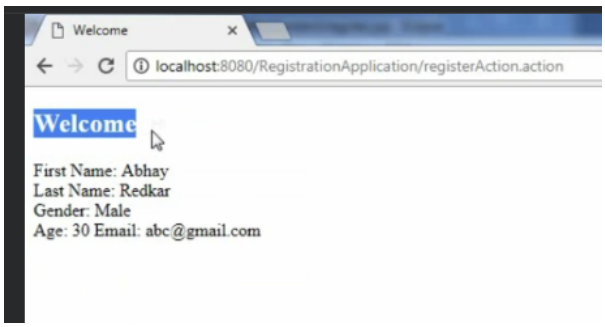
Different UI components in registration form - Registration jsp



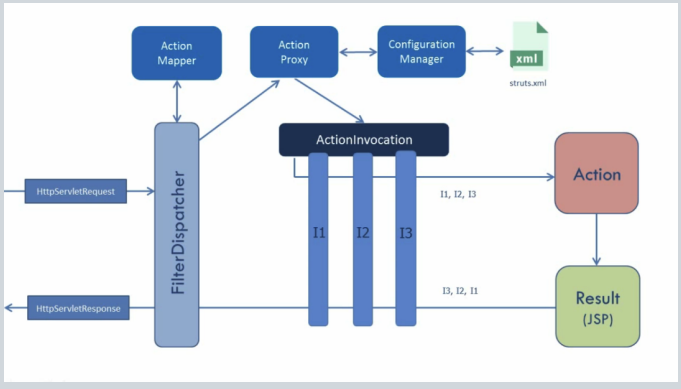
And welcome page







15. Struts 2 Architecture in detail

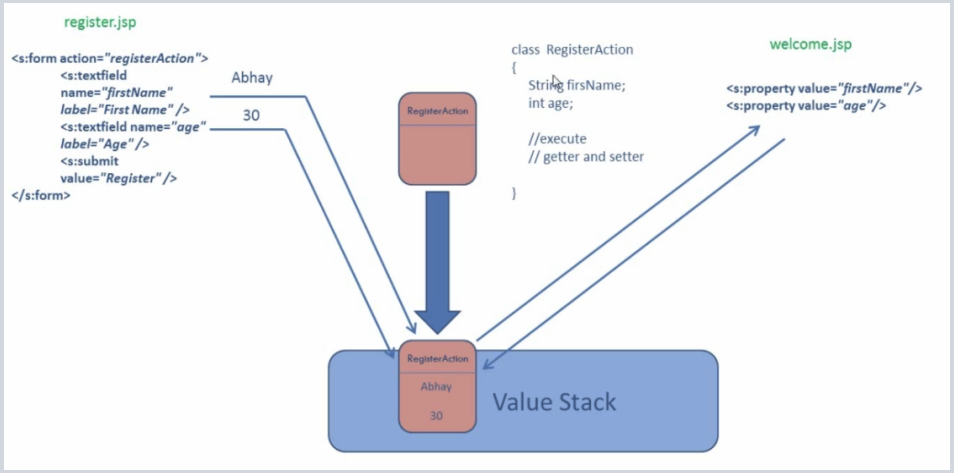


16. ValueStack and OGNL

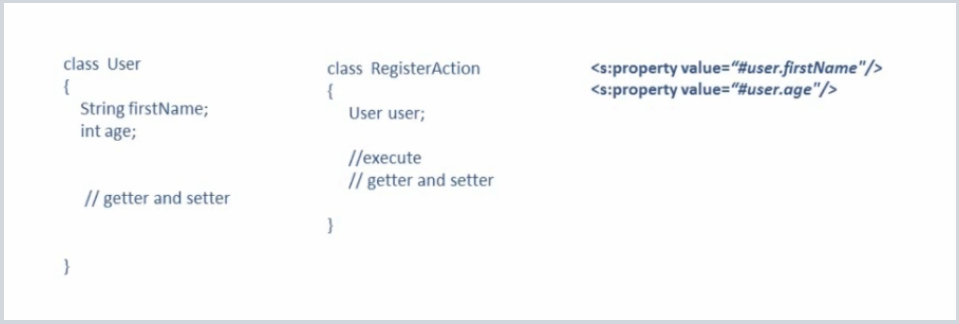
How values are going from jsp to action java and vice versa

- ValueStack is a storage area which holds all application specific data or data which is associated with the processing of a request.
- OGNL (Object-Graph Navigation Language) is an expression language which is used to access and manipulate data stored on the ValueStack.

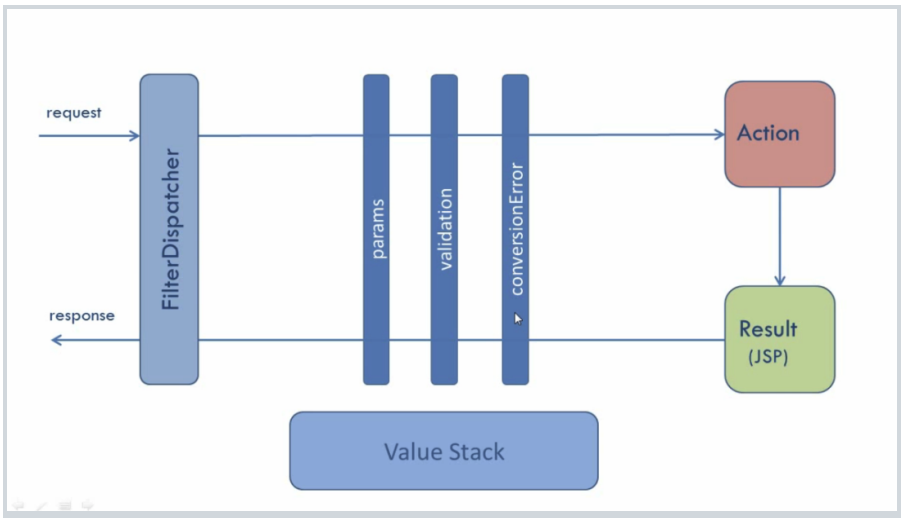
All values are stored in the value stack. Welcome jsp to get values from the value stack using OGNL. Notice everywhere we maintained same variable names.



In case of objects - OGNL helps in the below form.



17. Interceptors : Addresses the cross cutting concerts - which are not related to business.



register.jsp | welcome.jsp | *RegisterAction.java | struts.xml | web.xml

```java
import com.opensymphony.xwork2.ActionSupport;

public class RegisterAction extends ActionSupport{

    String firstName;
    String lastName;
    String gender;
    Integer age;
    String email;

    public String execute()
    {
        System.out.println("execute() method called");
        return "success";
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
```

Extend ActionSupport class.

And override validate method as shown below.

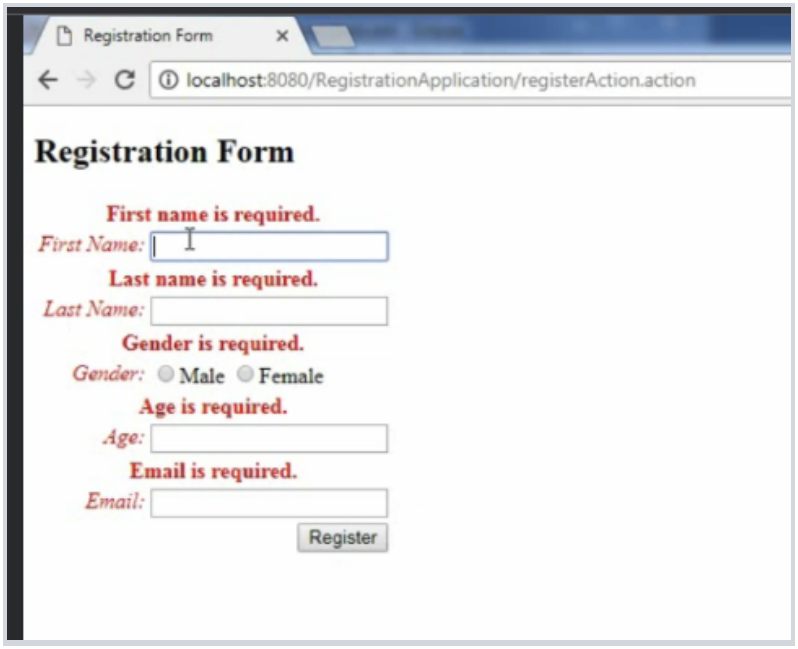
```java
public void validate()
{
    if (firstName.equals("")) {
        addFieldError("firstName", "First name is required.");
    }

    if (lastName.equals("")) {
        addFieldError("lastName", "Last name is required.");
    }

    if (gender == null) {
        addFieldError("gender", "Gender is required.");
    }
```

```
66
67          if (gender == null) {
68              addFieldError("gender", "Gender is required.");
69          }
70
71          if (age == null) {
72              addFieldError("age", "Age is required.");
73          }
74          else if(age <= 18)
75          {
76              addFieldError("age", "Age should be above 18.");
77          }
78
79          if (email.equals("")) {
80              addFieldError("email", "Email is required.");
81          }
```

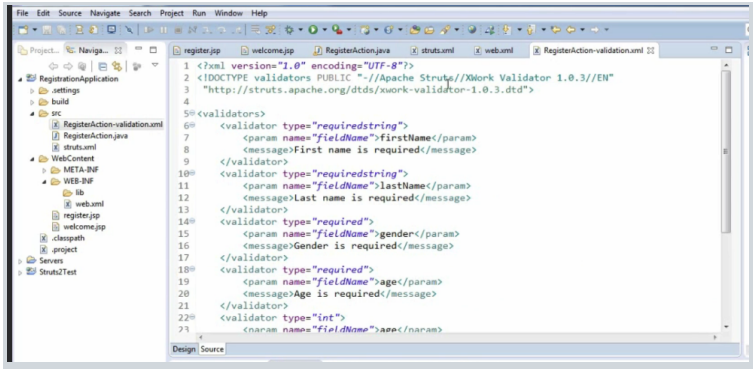
register.jsp | welcome.jsp | RegisterAction.java | *struts.xml | web.xml

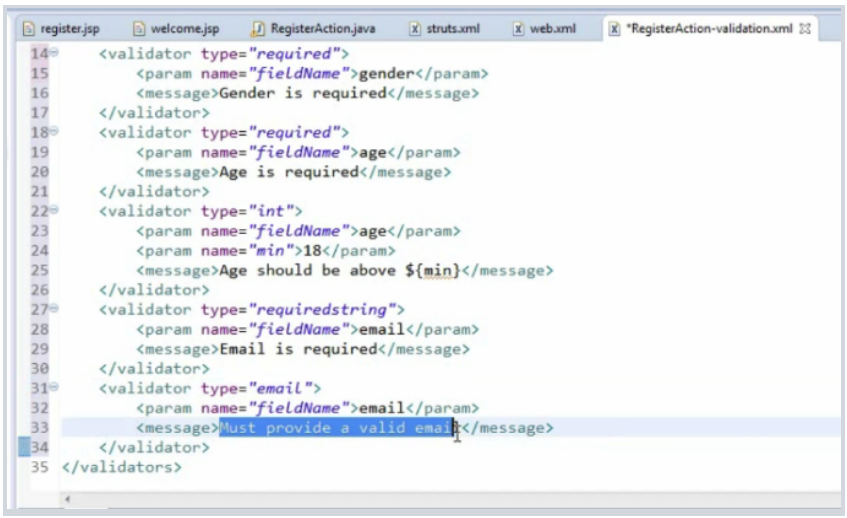
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE struts PUBLIC
3      "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
4      "http://struts.apache.org/dtds/struts-2.5.dtd">
5
6  <struts>
7      <package name="register" extends="struts-default">
8          <action name="registerAction" class="RegisterAction">
9              <result name="success">/welcome.jsp</result>
10             <result name="input">/register.jsp</result>
11         </action>
12     </package>
13 </struts>
```

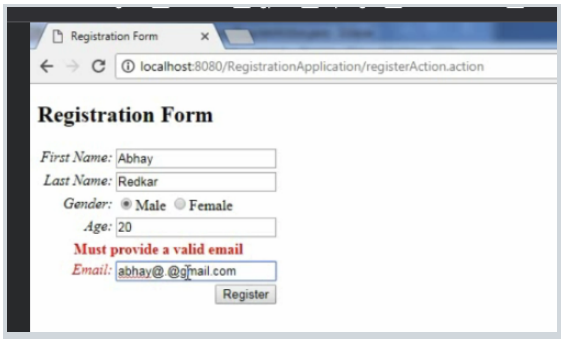
In case of any error page should be in same register.jsp only

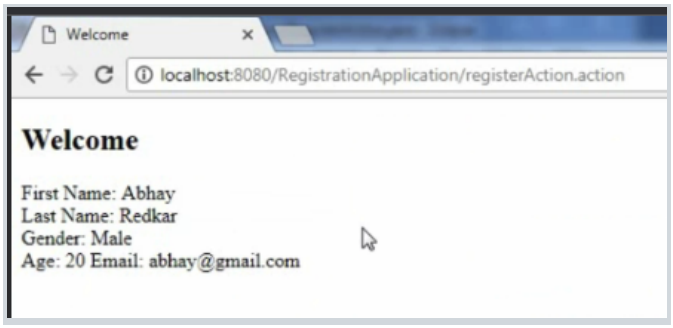


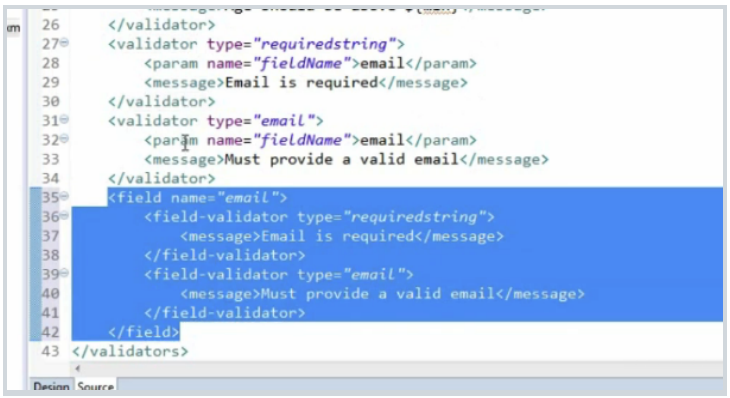
Something with validation xml. The file name must match with the action class name and followed by validation.xml as shown below.











Field validator will help to group multiple validators into groups as shown above. In the above case from line number 27 to 34 not required. Lines 35 to 42 covered indirectly.