Fokane Sakshi Anil
TE - A - 42

Unit No :- V

Q.2) Draw & explain neat diagram which depicts MVC to the Structs architecture

~> The architecture of struts is purely MVC (Model - View - Controller architecture) based.

- There are five core components in the MVC architecture of struts 2.

• Architecture Components:

1. Actions
2. Interceptors
3. Value Stack/OGNL
4. Result types
5. View technologies

- There is slight difference between Struts 2 & traditional MVC framework is the manner that the role of model is taken by action instead of controller.

- Fig 1-Q depicts the MVC architecture of Struts 2. The Model, View & Controller are implemented as follows:

• Model : Implemented with actions
• View : Implemented with combination of result types & result.
• Controller: Implemented with a struts 2 dispatch servlet filter & interceptors.

- The functionality of value stack and OGNL (object-Graph Navigation language) generally is to provide common thread, linking & supporting the integration in between different application components.

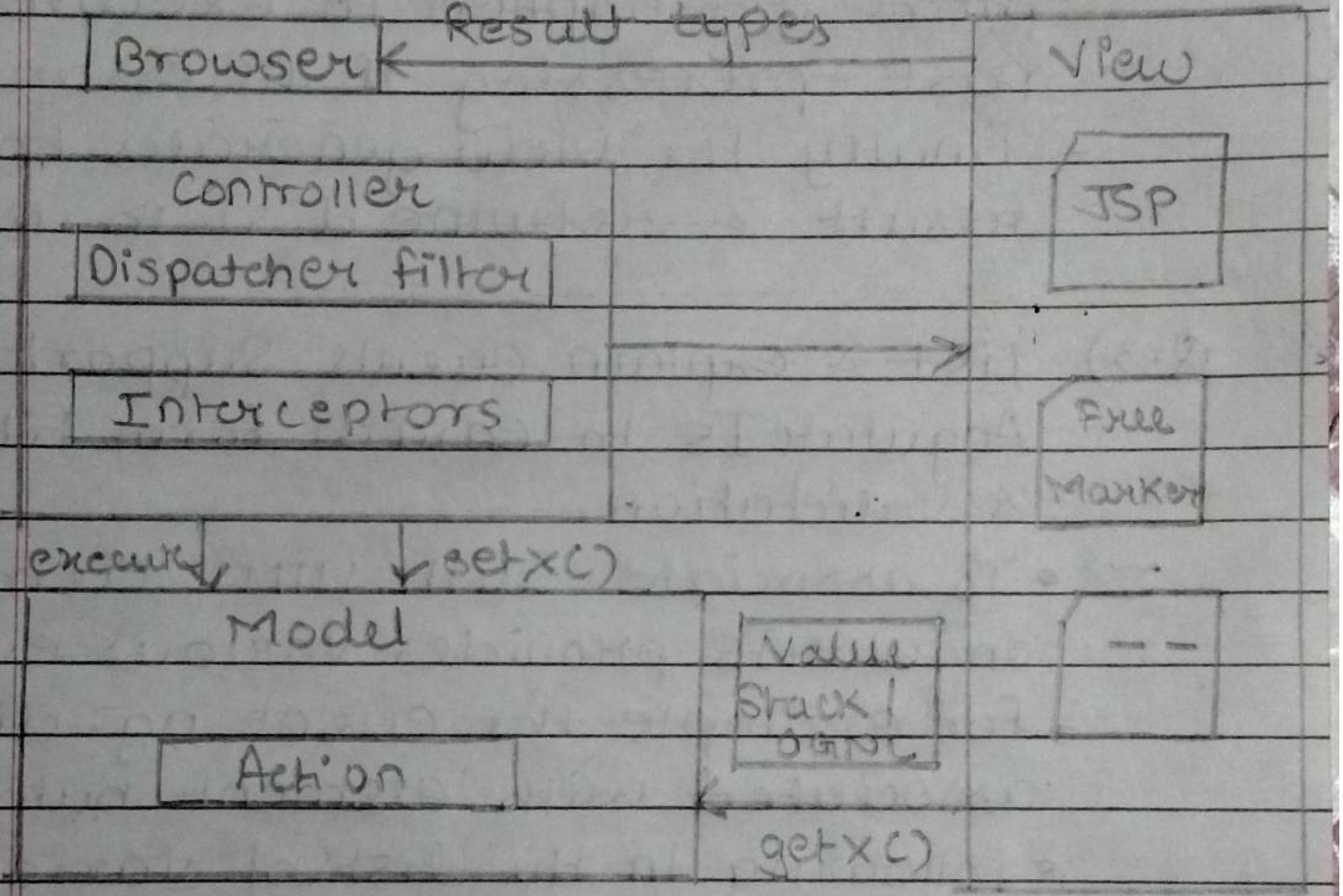

Fig : 1-Q. Struts Architecture

- Request life cycle :-
- Based on fig 1-Q the request life cycle in Struts 2 can be explained as follows:
- Request is sent by user to server for some resource.
- This request is accepted by the filterDispatcher & appropriate action is determined.

- functionalities of configured interceptors like validation, file upload are applied.
- To carry out the requested operat^n selected action is implemented.
- Another time, the configured interceptors are applied to execute any post-processing if necessary.
- Finally the view generates the result & returned it to user.

Q.3) List & explain events supported by AngularJS to enrich form filling & validation.

~> • To associate with HTML elements, angularJS provides various events. For example the event ng-click is associated with element button.

• following in the list of various events provided by AngularJS

1. ng-click
2. ng-dbl-click
3. ng-keydown
4. ng-keyup
5. ng-keypress
6. ng-change
7. ng-mousedown
8. ng-mouseup
9. ng-mouseenter
10. ng-mouseleav
11. ng-mousemove
12. ng-mouseover

eg:-

Resetting the form using on-click directive.

```
<!DOCTYPE html>
<html lang = "en">
<script.
src = "https://ajax.googleapis.com/ajax/libs
/angularjs/1.6.4/angular.min.js"></script>
<body>
<font size = 5>
<div ng-app = "mainApp" ng-controller =
"formController">
<form novalide>
First Name:<br>
<input type = "text" ng-model = "user.
first_Name"><br>
last Name:<br>
<input type = "text" ng-model = "user.
last_Name">
<br><br>
<button ng-click = "reset()">Reset</butt
</form>
```

Reset() function is called on click of button

```
<p> form = {{user}} </p>
<p> master = {{master}} </p>
</div>
```

Displays data of user & master (Default)

```
<script>
var app = angular.module("mainApp',[])
app.controller(`formController', function($
scope){
$scope.master = {first_Name: "Kunal", last
```

```
Name :"B");
$scope.reset = function(){
 $scope.user = angular.copy($scope.
 master);
```

copies data of master into user

```
};
 $scope.reset();
});
</script>
</font>
</body>
</html>
```

Q.4) What are different ways to create Service in AngularJS ? Explain Seperation of Concern in AngularJS.

→ • There are 4. different ways to declare a service.

(i) Registering a existing value as a Service.

(ii) Registering a factory function to create the Singleton Service instance.

(iii) Registering a constructor function to create the Singleton Service instance.

(iv) Registering a service factory which can be Configured.

- Separation of Concern :-

- Separation of concerns means that you have a well defined structure of your application: the data model in the application is decoupled from the business & presentation logic. It is the base of the MVC pattern, which defines the view.
- follows this guidance how to architect your AngularJS application.

- The model should:

1) Include the domain data;
2) Implement the management of the domain data.
3) Expose an API that makes possibly the model usage in controller or other service;

- The Model should not;

1) Provide the details on how the domai data is managed;
2) Contain logic that transforms the model based on user interaction;
3) Contain logic for displaying data to the user;

- A controller should;

1) contain the logic required to initialize the scope;
2) contain the logic required by the view to present data from scope;
3) Contain the logic required to update the scope based on user interaction.

- A Controller should not :

1) Contain logic that manipulates the DOM.

2) Contain logic that manages the persistence of data.

3) Manipulate data outside of scope

- A View Should :

1) Contain the logic & markup required to present data to the user.

- A View Should not :

1) contain complex logic.

2) Contain logic that Creates, Stores, or manipulates the domain model.

Q.5) write Short note on NodeJS.

~> 1) Node.js is an open source, cross platform JavaScript runtime environment for executing JavaScript code at Server Side.

2) Node.js enables Javascript to be used for Server Side Scripting & runs scripts server side to produce dynamic web page contents before the page is sent to user web browser.

3) Consequently, Node.js has become one of the foundational elements of the "JavaScript Everywhere" paradigm, allowing web application

development to unify around a single programming language.

- features of Node.js
1) Asynchronous & Event Driven
2) very Fast
3) Single Threaded but Highly Scalable
4) No Buffering
5) Free License.

1) Asynchronous & Event Driven :~
The API provided by Node.js are asynchronous, that means they are non-blocking. Hence the Server which is based on the Node.js does not have to wait for API to return data.
2) very fast :~
As the Node.js is built on JavaScript Engine of Google, chrome, its processing speed is very fast.
3) Single Threaded but Highly Scalable :-
The Model of Node.js is single threaded with event looping. The Event mechanism provides a way to the Server to respond in non-blocking manner.
4) No Buffering :-
Buffering is avoided in Node.js appⁿ
The data is Outputs in chunks
5) Free License :~
Node.js is released under the MIT license.