Spring MVC

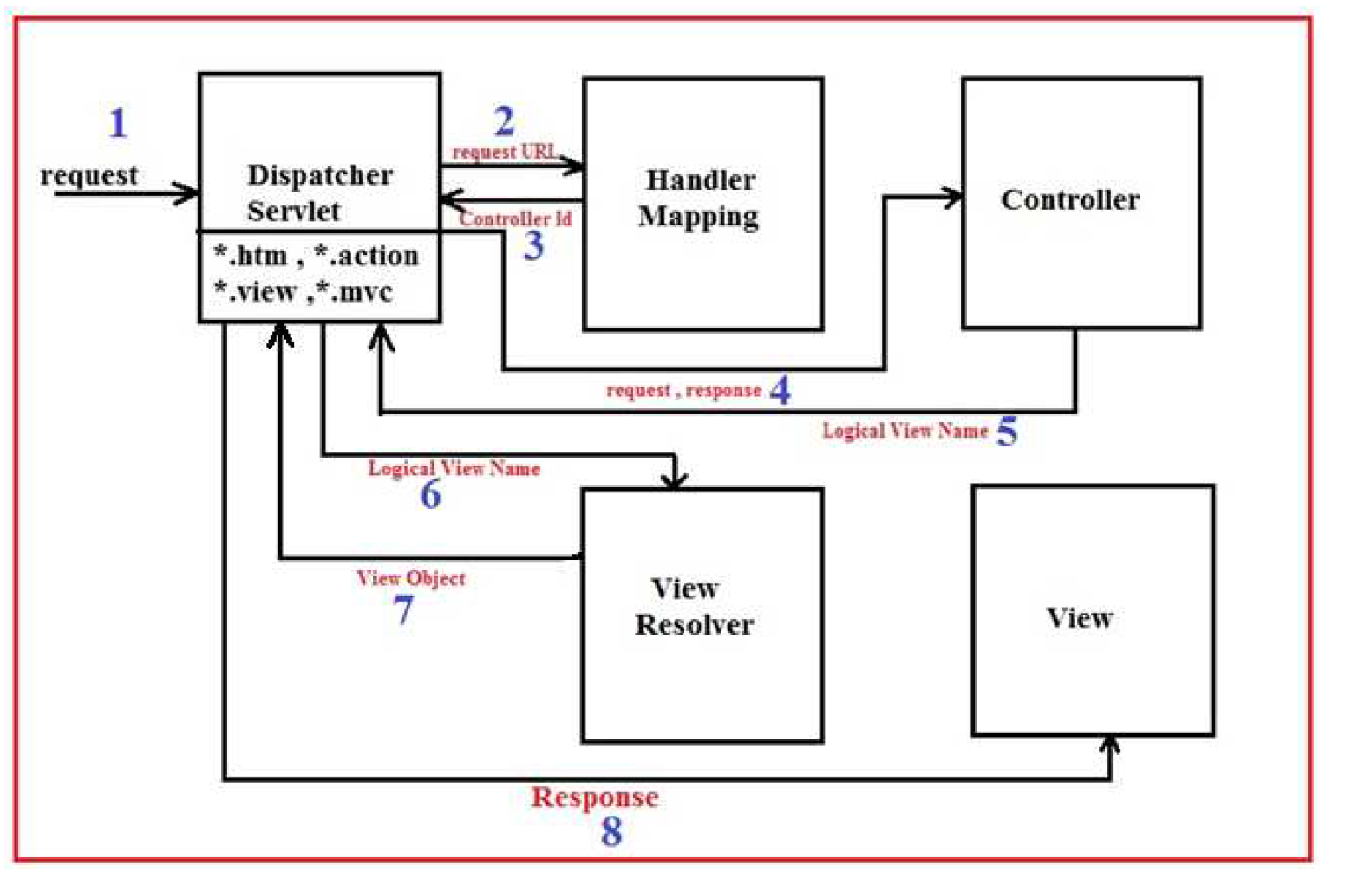
* Spring MVC is the module which use for developing web application.
* In this module we will see different features & advantages when compared with other web based frameworks.

Difference Between Spring and Other WebBased Frameworks.

* 1.Other Frameworks will always can render a view as JSP or html.
* Spring is limited to render view as jsp or html.Spring can render view as anything like pdf,excel.As they make view As an interface for that implementation could be anything.
* 2.In other web-based framework they made request wrapping process automated, but they fail in handling
* Attribute value/conversation.
* **Ex:** If one of the fields is taking int values but user has entered wrong input while mapping it will generate a big stack trace to the user, which user cannot understand?
* In **spring** if any attribute conversion problem is there then spring will not generate a big stacks trace rather it will render same page to user along with validation message to render correct data even it will reserve previous data also.
* 3.In struts to perform request wrapping programmer has to provide one VO class which contain all the attributes as String & attribute names & input field name should be same.
* Moreover programmer has to write struct-config.xml to tell to ActionServlet and that VO class extends from ActionForm class.
* Spring has one command class which help in spring user input in original form there is no need to type conversion spring has one great feature ie.properrtyEditer which automatically build corresponding values to particular attributes .
* 4.In struts validation logic written by programmer but it become boiler plate logic if there are multiple view available with same fields then for every view we have to write validation logic.Validation logic tight to each & every ActionFrom.
* **Spring** has provided command class there is no need to write any conversation logic rather command is the simple POJO class with original values we can directly use this class to perform business operations.
* 5. To apply common prepossessing logic or post processing logic functionality struts has provided filter as web component.
* Unlike struts, **Spring** has provide more advanced feature i.e. interceptors which apply common functionality.
* These Interceptor will not check in entry level these Interceptor are sit in spring MVC level and called along with controller class those are not called with servlet.
* So which controller you want to apply common prepossessing logic or post processing logic you can apply.
* So if url also change there is no effect of Interceptor.
* 6.In struts writing strut-config.xml is critical job because programmer has to write mapping configuration i.e. Form mapping as well as action mapping.
* If there is change in front URL then again we have to change struts-config.xml, maintenance cost more.
* Unlike struts, **Spring** doesn’t provided an configuration file, it’s automatic process in spring if there is change so there is no need change anything.

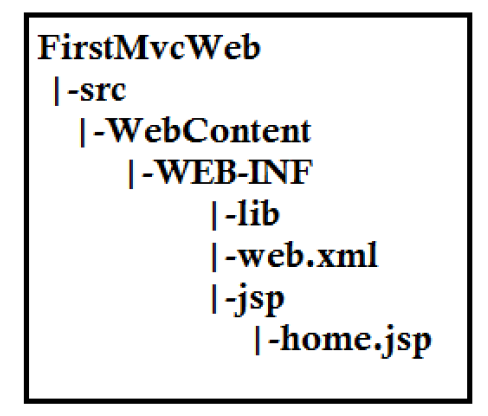
MVC Flow

* In case of Spring MVC user has to send the request with extension \*.htm or \*.mvc or \*.web or\*.action depends on the DispatcherServlet configuration wild card pattern.
* When user send the request it will be received by DispatcherServlet. Now DispatcherServlet asks the HandlerMapping to give the Controller information to whome he has to send the request by giving incoming url.



* Now HandlerMapping will take the url and map the appropriate Controller and return the Controller information to DispatcherServlet.
* Now DispatcherServlet will forward the request to Controller for processing the request. Now controller will process the request and return back the logical view name to the DispatcherServlet.
* After getting the logical view name DispatcherServlet will ask the ViewResolver to resolve the view by passing logical view name.
* View resolver will take the logical view and creates the object of the view which should be rendered by DispatcherServlet and return the View Object to DispatcherServlet.
* Now by getting the View Object DispatcherServlet will render the appropriate view.

MVC Project DirectStructure:



* In above project directory structure FirstMvcWeb is the root directory of our application.
* This application will show how to render the home.jsp.
* In a typical web application we put jsp pages in WebContent directory but in above application we have to put the home.jsp inside the WEB-INF.
* In case of Spring MVC it is recommended to keep View inside the WEB-INF.
* **what happens when we keep view in WebContent directory in case of Servlet and Jsp or Struts framework**

**problem-1**

* WebContent is a public directory.
* Views in this directory can be accessed by any users directly.
* In this case our presentation layer technology will get exposed to End User.

Problem 2

* As End-User can directly access the view he or she may send the request to access the view which expect some data as input.
* In this case user will get an ugly error message which will give bad experience of using our application to the user.

Problem 3

* If users of our application is technically strong then they might hack our application by using session and knowing the type of view (jsp/html).
* **Advantage of placing the view under WEB-INF folder**
* In case of Servlet and JSP or Struts or any other framework except spring MVC views are either JSP or Html.
* There is no point in placing the views in WEB-INF to abstract it from user.
* That’s why it is not recommended to place the views inside WEB-INF.
* In case of Spring MVC views may be anything like jsp, html, xml, pdf, Excel Sheet etc.
* So, due to above reasons it is recommended to place the views inside WEB-INF directory only.
* WEB-INF is the protected directory in the project directory structure.
* Anything placed inside the WEB-INF directory can be used by application component only.
* It cannot be used by user directly from outside.
* In this way our views will be abstracted from end user .