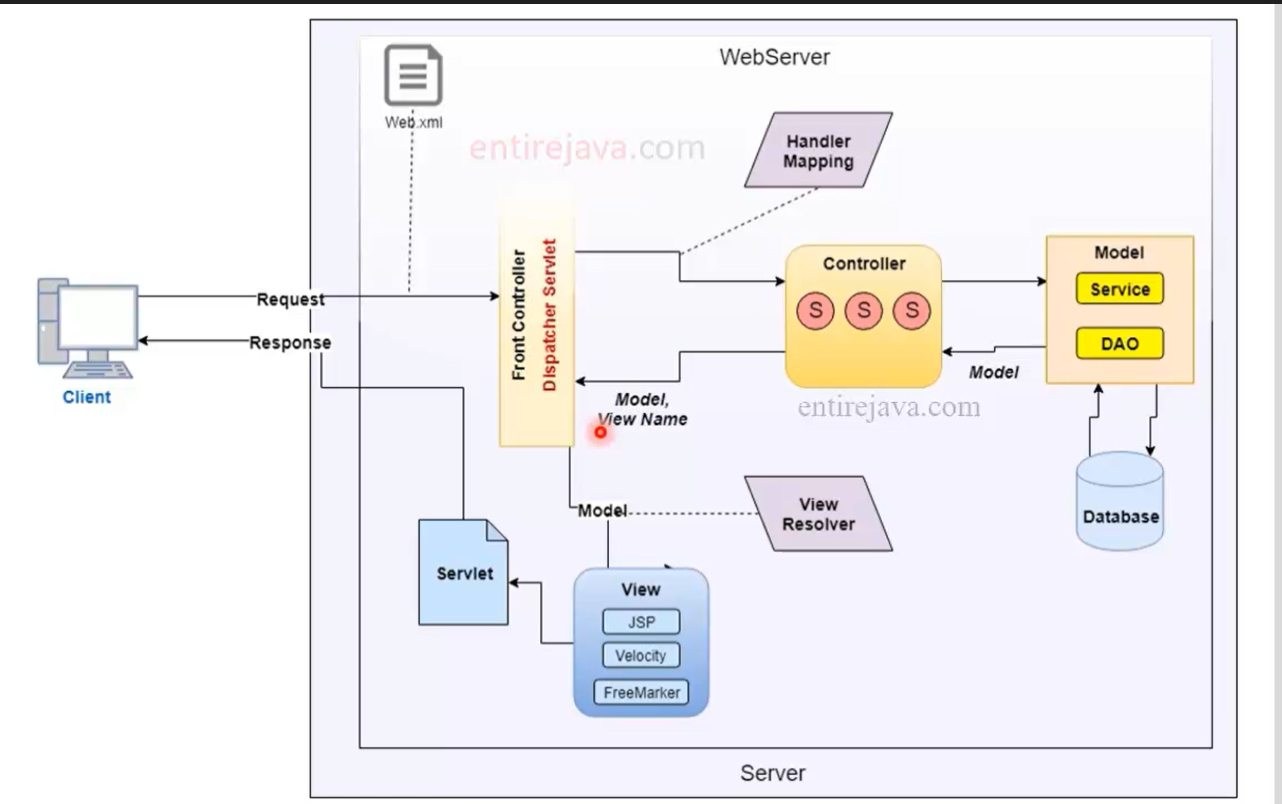
**Spring MVC**

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications.

* The **Model** encapsulates the application data and in general, they will consist of POJO.
* The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
* The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.

A *DispatcherServlet* that handles all the HTTP requests and responses.





The difference between traditional MVC and Spring MVC is Spring I.e., Dispatcher Servlet. When an HTTP request is received from the Client. It will go to the Web.XML file and tries to look for a mapping. Here in Spring MVC, we will map to Front Controller not directly to the servlets. *DispatcherServlet* consults the *HandlerMapping* to call the appropriate *Controller*. We need to explicitly specify the URL mapping inside our config file. But in with annotations we don’t need to specify URL. The *Controller* takes the request and calls the appropriate service methods based on the used GET or POST method. The service method will set model data based on defined business logic. The Service layer communicates with DAO Objects or Data Access Object to able to communicate with the Database. Then we will get the Result, In Spring terms results are called as Model. The Model is known as an Object that holds the data. The servlet will send the model to the end-user along with the View name( doesn’t send any data like where it is residing or file extension) to the Front Controller. The Front Controller will send the model object to the View with the help of *ViewResolver* to pick up the defined view for the request which is configured in the Spring Config file. Then it will convert the file into servlets and sends the response to the Client.

The **web.xml** file will be kept in the WebContent/WEB-INF directory of your web application

* The *[servlet-name]-servlet.xml* file will be used to create the beans defined, overriding the definitions of any beans defined with the same name in the global scope.
* The *<context:component-scan...>* tag will be used to activate Spring MVC annotation scanning capability which allows to make use of annotations like @Controller and @RequestMapping etc.
* The *InternalResourceViewResolver* will have rules defined to resolve the view names. As per the above-defined rule, a logical view named **hello** is delegated to a view implementation located at */WEB-INF/jsp/hello.jsp* .

The **@Controller** annotation indicates that a particular class serves the role of a controller.

The **@RequestMapping** annotation is used to map a URL to either an entire class or a particular handler method.