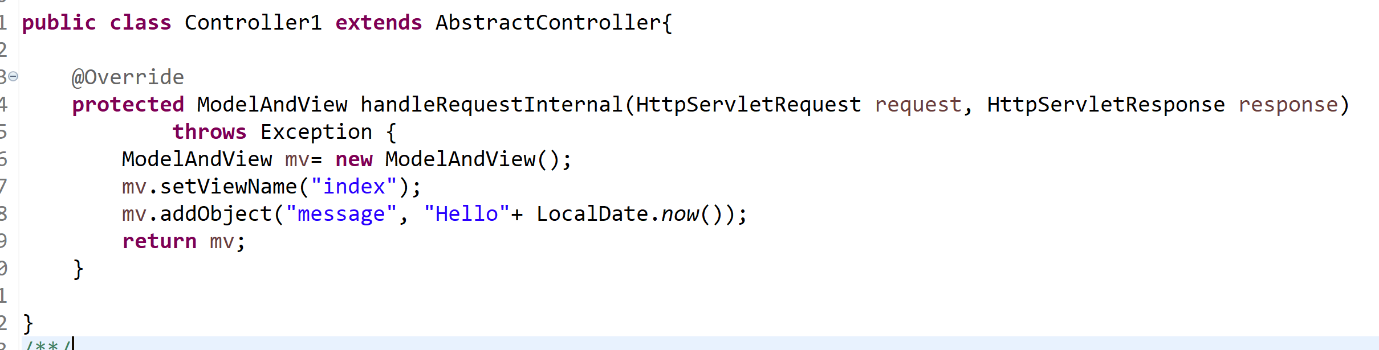
Spring MVC

Dispatcher Servlet : Front Controller

Request -> web.xml -> web container(jsp/ html/css)/ spring container : dynamic web page-> DispatcherServlet -> spring config -> controller (HandlerMapping) -> set ur Model/data, returns the view name -> ViewResolver -> View page -> display the view

Login.html : username, password -> Controller (validate) -> Welcome page / Error Page

1. Controller Interface
2. AbstractController class: BeanNameUrlHandlerMapping



1. @Controller: RequestMappingHandlerMapping

By-default, dispatcher servlet looks for a config file with a name <servletname>-servlet.xml. But if we want it to read some other file, we can configure the param contextConfigLocation as a init param.

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/springconfig.xml</param-value>

</init-param>

In this case it will read springconfig.xml instead.

If we want to configure more than one config file,

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/springconfig.xml,/WEB-INF/spring-dao.xml</param-value>

</init-param>

If we want spring to read java config instead of xml config,

<init-param>

<param-name>contextClass</param-name>

<param-value>org.springframework.web.context.support.AnnotationConfigWebApplicationContext</param-value>

</init-param>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>com.java.SpringConfig</param-value>

</init-param>

**Application**

Login.html -> enter username & password -> Correct -> address, phNo, Gender(radio button), subjects (java, j2ee, )..checkbox, tel No -> submit -> display all details except password.

* Wrong credentials -> Invalid details.

Spring jdbc : module for integrating jdbc

User class: username, pwd, address, phNo, gender..

**Annotations:**

**Spring-jdbc :** Module of spring which makes integration with jdbc very easy.

JDBCTemplate: lot of methods for querying: querForInt(), it manages for u session t, tx automatically. Every ddl statement in this template is auto-commit.

Jdbc: executeQuery(): ResultSet

Update(): for insert, update, delete

execute(): for ddl

**Interceptor:**

Spring:

Request -> handlerMapping -> Controller -> viewname -> ViewResolver -> view -> response

preHandle postHandle afterCompletion