

ResponseBody and View Resolver

If you want to write some business login then java is the right place but if you want to do some webpage stuff to show information on browser then you can use a JSP pages or templates like ftl. We can create JSP's in Spring boot application in the folder /src/main/webapp/WEB-INF/jsp/login.jsp

Create a source folder called src/main/webapp, WEB-INF folder inside and jsp folder inside

```
login.jsp
<html>
    <head>
        <title>First web application</title>
    </head>

    <body>
        My first JSP !!!
    </body>
</html>
```

```
LoginController
@Controller
public class LoginController {

    @RequestMapping("/login")
    @ResponseBody
    public String loginMessage() {
        return "login";
    }
}
```

To make your application to look for a view you need to remove ResponseBody annotation.

```
@Controller
public class LoginController {

    @RequestMapping("/login")
    public String loginMessage() {
        return "login";
    }
}
```

Then DispatcherServlet looks for a view and again gives white label error page this is because DispatcherServlet doesn't know the path of the JSP page. Then how do tell DispatcherServlet where the login.jsp page is located ? That's where concept of View Resolver comes into the picture. Whenever we provide a view name to DispatcherServlet it uses something called as view resolver to resolve it. So we can make this mapping
login -> /src/main/webapp/WEB-INF/jsp/login.jsp

View Resolver

This can be specified in the application.properties file as
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp

But again it gives white label error page this is because our embedded tomcat server doesn't have support for JSP pages. To enable the support we need to add dependency called tomcat-embed-jasper

```
<dependency>  
    <groupId>org.apache.tomcat.embed</groupId>  
    <artifactId>tomcat-embed-jasper</artifactId>  
</dependency>
```

Whenever we add a new dependency we need to stop and start the entire server.

Access <http://localhost:8080/login> on browser it gives jsp page as response

My first JSP !!!