

Spring MVC Controller

Create a web application and run it.

After the application starts goto browser and access `http://localhost:8080/login`

It gives response as Whitelabel Error Page (default error page).

This is because we have not created anything at `/login` so it says application has no explicit mapping.

If it is a JSP/Spring MVC application the error message would be like page not found.

Now how do you map your requests to Spring boot application ?

Controller

Now create a `LoginController`, create a method inside the controller class and save it. You can see that application gets shutdown and restart automatically this is because of the DevTools dependency in our spring boot application.

```
public class LoginController {  
  
    @RequestMapping("/login")  
    public String loginMessage() {  
        return "Hello World";  
    }  
}
```

Now you won't find mapping in the console logs that's because this particular class is not being picked up by the spring. So to make your spring application to pick up the class you need to annotate it with `Controller`

```
@Controller  
public class LoginController {  
  
    @RequestMapping("/login")  
    public String loginMessage() {  
        return "Hello World";  
    }  
}
```

Now to know what's going on in this spring boot application we can enable logging level. If we enable logging level to `DEBUG` it prints lots of log messages. To log only specified package we can mention it in `application.properties`

```
logging.level.org.springframework.web=DEBUG
```

Even now it gives response as Whitelabel Error Page. That's because `DispatcherServlet` will do the mapping i.e `/login` is mapped to `LoginController.loginMessage`. Once this method returns text message `Hello World` `DispatcherServlet` looks for the view named `Hello World`. Since it cannot find it it gives Whitelabel Error Page. View is something like JSP page.

```

2021-01-01 02:53:17.018 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : GET "/login",
parameters={}
2021-01-01 02:53:17.019 DEBUG 8606 --- [nio-8080-exec-1]
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to
com.example.demo.controller.LoginController#loginMessage()
2021-01-01 02:53:17.023 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.s.v.ContentNegotiatingViewResolver : Selected 'text/html'
given [text/html, application/xhtml+xml, image/webp, application/
xml;q=0.9, */*;q=0.8]
2021-01-01 02:53:17.023 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.servlet.view.InternalResourceView : View name 'Hello
World !!!', model {}
2021-01-01 02:53:17.025 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.servlet.view.InternalResourceView : Forwarding to [Hello
World !!!]
2021-01-01 02:53:17.028 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : "FORWARD" dispatch for
GET "/Hello World !!!", parameters={}
2021-01-01 02:53:17.033 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped to
ResourceHttpRequestHandler ["classpath:/META-INF/resources/",
"classpath:/resources/", "classpath:/static/", "classpath:/public/",
"/"]
2021-01-01 02:53:17.035 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.s.r.ResourceHttpRequestHandler    : Resource not found
2021-01-01 02:53:17.035 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : Exiting from "FORWARD"
dispatch, status 404
2021-01-01 02:53:17.036 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : Completed 404 NOT_FOUND
2021-01-01 02:53:17.036 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : "ERROR" dispatch for GET
"/error", parameters={}
2021-01-01 02:53:17.037 DEBUG 8606 --- [nio-8080-exec-1]
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped to
org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorC
ontroller#errorHtml(HttpServletRequest, HttpServletResponse)
2021-01-01 02:53:17.053 DEBUG 8606 --- [nio-8080-exec-1]
o.s.w.s.v.ContentNegotiatingViewResolver : Selected 'text/html'
given [text/html, text/html;q=0.8]
2021-01-01 02:53:17.057 DEBUG 8606 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet      : Exiting from "ERROR"
dispatch, status 404

```

```

@Controller
public class LoginController {

    @RequestMapping("/login")
    @ResponseBody
    public String loginMessage() {
        return "Hello World !!!";
    }
}

```

```
}
```

So if you don't want your DispatcherServlet to look for a view then you can use the annotation called as ResponseBody. When you use this annotation DispatcherServlet uses some message converters to give back the response to the browser.

Access <http://localhost:8080/login> on browser it gives response as Hello World
Modify Hello World to Hello World!!! and access browser it gives response as Hello World!!!