

SPRING ASSIGNMENT Day_5&6

Harshit Kushmakar| 16896

1. Create the First Spring MVC application to print “Hello World” on the JSP page. Create “index.jsp” with hyperlink to the Controller and return a JSP Page as response which prints “Hello World”
Make the application first with XML configuration and then with Java configuration.

day5assignment/ conf/
SpringConfig1.java

```
package day5assignment.conf;

import org.springframework.context.annotation.Bean; import
org.springframework.context.annotation.ComponentScan;
import
org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.EnableWebMvc;
import
org.springframework.web.servlet.view.InternalResourceViewResolver
;

@Configuration
@ComponentScan("day5assignment") //they will be automatically
findout
@EnableWebMvc // WebApplicationAnnotationConfigContext
public class SpringConfig1 {
    @Bean
    public InternalResourceViewResolver getResolver() {
        InternalResourceViewResolver resolver = new
InternalResourceViewResolver();
        resolver.setPrefix("/");
        resolver.setSuffix(".jsp");          return
        resolver;
    }
}
```

day5assignment/ conf/
AppInitializer1.java

```
package day5assignment.conf;

import
org.springframework.web.servlet.support.AbstractAnnotationConfigD
ispatcherServletInitializer;
    public class AppInitializer1
extends
```

```

AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
    return new Class[] { SpringConfig1.class };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
    return null;
    }

    @Override
    protected String[] getServletMappings() {
    return new String[] { "*.do", "*.htm" };
    }
}

```

webapp/ day5assignment/ index.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix = "spring" uri
="http://www.springframework.org/tags"%>
<html>
<head>
<title>Spring Assignment Day 5 & 6</title>
</head>
<body>
<h1>Question 1 </h1>
<a href="helloWorld.do"> Display </a>
<br>
</body>
</html>

```

day5assignment/ controller/ HelloWorldController.java

```

package day5assignment.controller;
import
org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
@Controller
public class HelloWorldController {

    @RequestMapping("/helloWorld.do")    public
String printHello(Model model) {
model.addAttribute("message", "Hello World");
return "day5assignment/HelloWorld";
    }

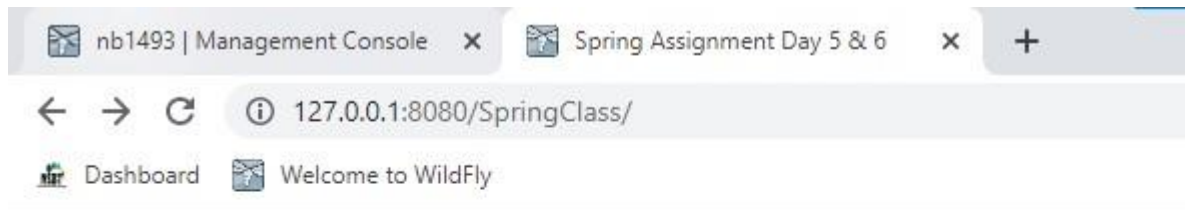
}

```

webapp/ day5assignment/ HelloWorld.jsp

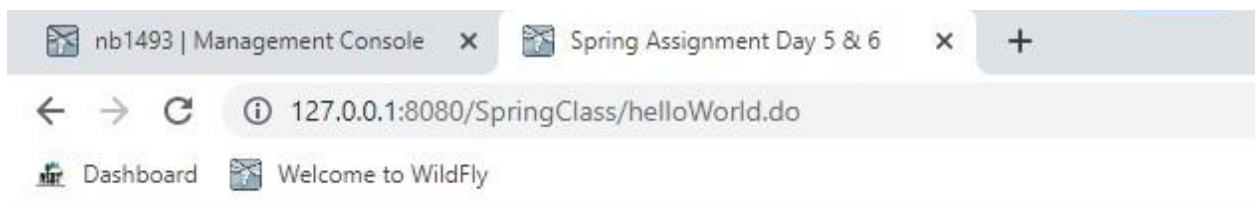
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<title>Spring Assignment Day 5 & 6</title>
</head>
<body>
<h1>Displaying data</h1>
<h2><%= request.getAttribute("message") %></h2>
</body>
</html>
```

Output:



Question 1

[Display](#)



Displaying data

Hello World

2. Change the above application by sending the name of the User as a Query String parameter to the link created in "index.jsp" page. Retrieve this name with the help of @RequestParam annotation

and then send it to the display.jsp page for printing. The response must look like as mentioned below:

Hello < username > Example:

Index.jsp contains the below hyperlink:

` Say Hello </ a>` The

output which need to be printed on display.jsp is:

Hello Ajay

webapp/ day5assignment/

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix = "spring" uri
="http://www.springframework.org/tags"%>
<html>
<head>
<title>Spring Assignment Day 5 & 6</title>
</head>
<body>
<h1>Question 1 </h1>
<a href="helloWorld.do"> Display </a>
<br>

<h1>Question 2 </h1>
<a href="sayHello.do?username=Ajay"> Display </a> <br>

</body>
</html>
```

day5assignment/

controller/

HelloWorldController.java

```
package day5assignment.controller;

import org.springframework.stereotype.Controller; import
org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam; import
org.springframework.web.servlet.ModelAndView;

@Controller
public class HelloWorldController {
```

```
// 2
```

```

        @RequestMapping(path="/sayHello.do", method = RequestMethod.GET)
        public ModelAndView add(@RequestParam("username") String username)
        {

            ModelAndView mv= new ModelAndView();
            mv.setViewName("day5assignment/display");
            mv.addObject("username", username);           return
            mv;

        }
    }
}

```

webapp/ day5assignment/ display.jsp

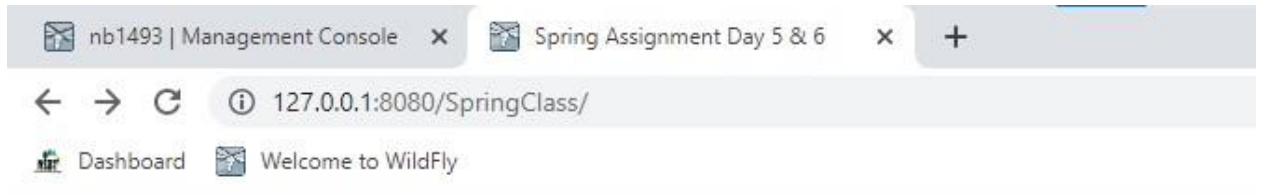
```

<%@page contentType="text/html" pageEncoding="UTF-8"%> <html>
<head>
<title>Displaying data in Session</title>
</head>
<body>
<h1>Displaying data </h1>
<br>

    <h2>
        Hello ${username}
    </h2>
</body>
</html>

```

Output:

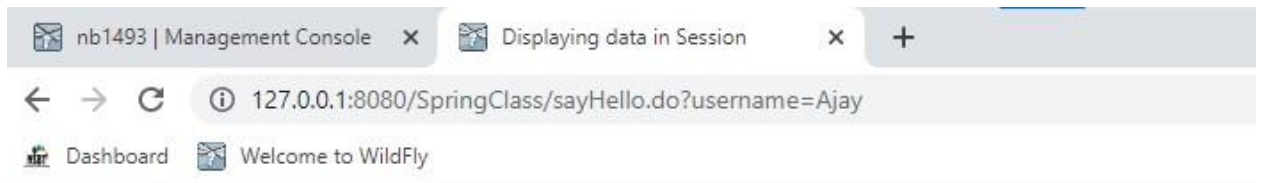


Question 1

[Display](#)

Question 2

[Display](#)



Displaying data

Hello Ajay

3. Change the above application to return the response with the help of ModelAndView object.
4. In the above application, check for the difference between forward and redirect. How will you do the same and what will be the impact. Maintain the difference in a readMe.txt file.

9 Forward

webapp/ day5assignment/
index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```

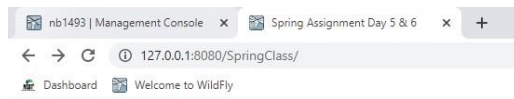
<%@ taglib prefix = "spring" uri
="http://www.springframework.org/tags"%>
<html>
<head>
<title>Spring Assignment Day 5 & 6</title>
</head>
<body>
<h1>Question 1 </h1>
<a href="helloWorld.do"> Display </a>
<br>

<h1>Question 2 </h1>
<a href="sayHello.do?username=Ajay"> Display </a>
<br>

<h1>Question 4 </h1>
<a href="question4.do"> Display </a>
<br>

</body>
</html>

```



Question 1

[Display](#)

Question 2

[Display](#)

Question 4

[Display](#)

day5assignment/ controller/ HelloWorldController.java

```

package day5assignment.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class HelloWorldController {

```

```

    @RequestMapping("/question4")
    public String sayHello(Model map) {
        map.addAttribute("msg", "Welcome to Spring using Model");

        //        return "redirect:day5assignment/question4.jsp";
        return "forward:day5assignment/question4.jsp";
    }
}

```

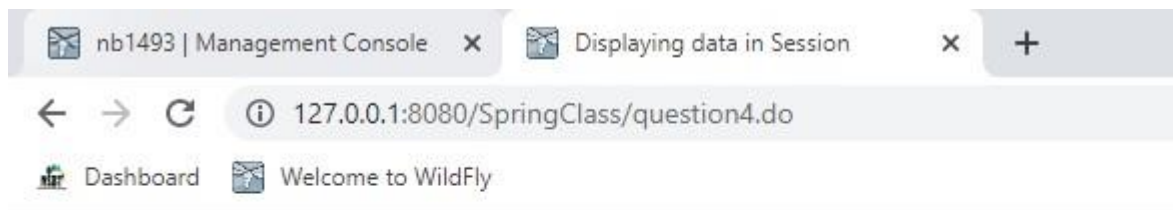
webapp/ day5assignment/ question4.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<title>Displaying data in Session</title>
</head>
<body>
<h1>Displaying data </h1>
Message is : <%= request.getAttribute("msg") %>

</body>
</html>

```



Displaying data

Message is : Welcome to Spring using Model

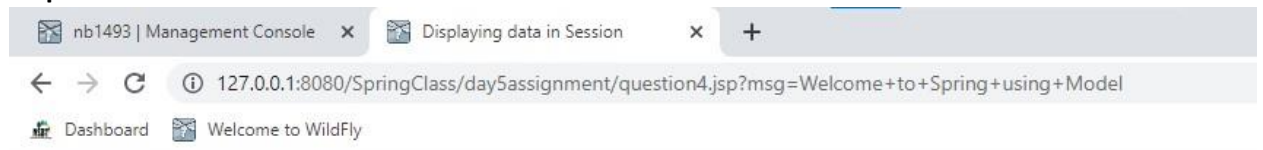
⑨ Redirect


```

34 // 4
35
36
37 @RequestMapping("/question4")
38 public String sayHello(Model map) {
39     map.addAttribute("msg", "Welcome to Spring using Model");
40
41     return "redirect:day5assignment/question4.jsp";
42     // return "forward:day5assignment/question4.jsp";
43
44     // Forward: is faster, the client browser is not involved, the request is transferred to the forwarded URL.
45     // Redirect: is slower, the client browser is involved, it creates a new request to the redirected URL.
46
47     // We passed the parameter "msg" with a value in both cases. Forwarded requests still carry this value, but redirected requests don't.
48 }
49
50
51

```

Output:



Displaying data

Message is : null

5. In the above application, create another controller to access Request Headers with the help of `@RequestHeader` annotation. Display different response based on the value of 'referrer' request header. If header value is null, it should take the user to error page else to home page.

webapp/ day5assignment/ index.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix = "spring" uri
="http://www.springframework.org/tags"%>
<html>
<head>
<title>Spring Assignment Day 5 & 6</title>
</head>
<body>
<h1>Question 1 </h1>
<a href="helloWorld.do"> Display </a>
<br>

<h1>Question 2 </h1>
<a href="sayHello.do?username=Ajay"> Display </a>
<br>

<h1>Question 3 </h1>
<a href="helloWorld.do"> Display </a>

```

```

<br>

<h1>Question 4 </h1>
<a href="question4.do"> Display </a>
<br>

<h1>Question 5 </h1>
<a href="question5.do"> Display </a>
<br><br>
Referer is : <%= request.getAttribute("referer")%>

</body>
</html>

```

day5assignment/
controller/
HelloWorldController.java

```

package day5assignment.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

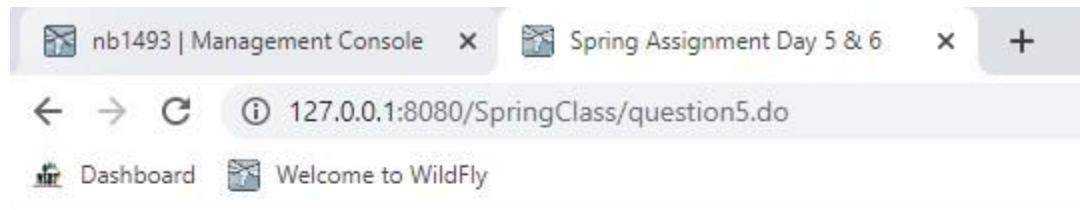
@Controller
public class HelloWorldController {

    // 5
    @RequestMapping("/question5")
    public ModelAndView requestHeader(@RequestHeader(value =
"Referer") String referer){

        ModelAndView mv = new ModelAndView();
        // referer= null;
        if(referer != null){
            mv.setViewName("day5assignment/question5");
            mv.addObject("referer", referer);
        }
        else {
            mv.setViewName("error");
        }
        return mv;
    }
}

```

Output:



Question 1

[Display](#)

Question 2

[Display](#)

Question 3

[Display](#)

Question 4

[Display](#)

Question 5

[Display](#)

Referer is : <http://127.0.0.1:8080/SpringClass/>

When referer = null;

error.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<title>error wala</title>
</head>
```

```

<body>
<h1>Displaying data </h1>

<h2>Error Name <%= request.getAttribute("name") %></h2>
<h2>Error Handler <%= request.getAttribute("errorHandler") %></h2>
<br>

</body>
</html>

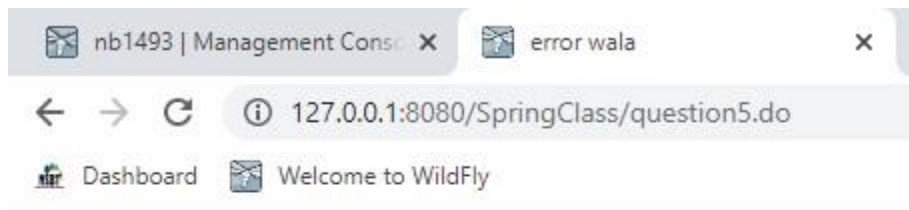
```

ExceptionHandler.java

```

package spring.controller;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.HandlerExceptionResolver;
import org.springframework.web.servlet.ModelAndView; import
javax.servlet.http.HttpServletRequest; import
javax.servlet.http.HttpServletResponse;
/**
 * Exception handler by implementing the interface
 * provided.
 * This will be the default error handler * and
 * override all other error handlers.
 *
 *
 */
@Component
public class ExceptionHandler implements HandlerExceptionResolver {
    @Override
    public ModelAndView resolveException(HttpServletRequest request,
                                       HttpServletResponse response,
                                       Object handler, Exception error) {
        //Log
        //we can do anything what we wish to do before sending error
        message to the user.
        System.out.println("Exception Handler: " + handler);
        System.out.println(error);
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("error");
        modelAndView.addObject("name", "Class Exception Handler");
        modelAndView.addObject("errorHandler", error);        return
        modelAndView;
    }
}

```



Displaying data

Error Name null

Error Handler null