

Spring MVC Tutorial (XML Config.)

Hello World Example

I will show you a Gradle + Spring 4 MVC, Hello World Example (JSP view), XML configuration.

Technologies used :

1. Gradle 2.0
2. Spring 4.1.6.RELEASE
3. Eclipse 4.4
4. JDK 1.7
5. Logback 1.1.3
6. Bootstrap 3

1. Project Structure

Download the project [source code](#) and review the project folder structure :

2. Gradle Build

2.1 Review the **build.gradle** file, this should be self-explanatory.



Build.gradle

```
apply plugin: 'java'
apply plugin: 'war'
apply plugin: 'eclipse-wtp'
apply plugin: 'jetty'

// JDK 7
sourceCompatibility = 1.7
targetCompatibility = 1.7

repositories {
    mavenLocal()
    mavenCentral()
}

dependencies {
    compile 'ch.qos.logback:logback-classic:1.1.3'
    compile 'org.springframework:spring-webmvc:4.1.6.RELEASE'
    compile 'javax.servlet:jstl:1.2'
}

// Embedded Jetty for testing
jettyRun{
    contextPath = "spring4"
    httpPort = 8080
}
```

```

}

jettyRunWar{
    contextPath = "spring4"
    httpPort = 8080
}

//For Eclipse IDE only
eclipse {

    wtp {
        component {

            //define context path, default to project folder name
            contextPath = 'spring4'

        }
    }
}
}

```

- 2.2 To make this project supports Eclipse IDE, issues **gradle eclipse** :

```
your-project$ gradle eclipse
```

3. Spring MVC

Spring MVC related stuff.

- 3.1 Spring Controller – **@Controller** and **@RequestMapping**.

WelcomeController.java

```

package com.mkyong.helloworld.web;

import java.util.Map;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.mkyong.helloworld.service.HelloWorldService;

@Controller
public class WelcomeController {

    private final Logger logger =
LoggerFactory.getLogger(WelcomeController.class);
    private final HelloWorldService helloWorldService;

    @Autowired
    public WelcomeController(HelloWorldService helloWorldService) {

```

```

        this.helloWorldService = helloWorldService;
    }

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String index(Map<String, Object> model) {

        logger.debug("index() is executed!");

        model.put("title", helloWorldService.getTitle(""));
        model.put("msg", helloWorldService.getDesc());

        return "index";
    }

    @RequestMapping(value = "/hello/{name:.+}", method = RequestMethod.GET)
    public ModelAndView hello(@PathVariable("name") String name) {

        logger.debug("hello() is executed - $name {}", name);

        ModelAndView model = new ModelAndView();
        model.setViewName("index");

        model.addObject("title", helloWorldService.getTitle(name));
        model.addObject("msg", helloWorldService.getDesc());

        return model;
    }
}

```

3.2 A service to generate a message.

HelloWorldService.java

```

package com.mkyong.helloworld.service;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Service;
import org.springframework.util.StringUtils;

@Service
public class HelloWorldService {

    private static final Logger logger =
        LoggerFactory.getLogger(HelloWorldService.class);

    public String getDesc() {

        logger.debug("getDesc() is executed!");

        return "Gradle + Spring MVC Hello World Example";
    }
}

```

```

public String getTitle(String name) {

    logger.debug("getTitle() is executed! $name : {}", name);

    if(StringUtils.isEmpty(name)){
        return "Hello World";
    }else{
        return "Hello " + name;
    }

}

}

```

3.3 Views – JSP + JSTL + bootstrap. A simple JSP page to display the model, and includes the static resources like css and js.

/WEB-INF/views/jsp/index.jsp

```

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html lang="en">
<head>
<title>Gradle + Spring MVC</title>

<spring:url value="/resources/core/css/hello.css" var="coreCss" />
<spring:url value="/resources/core/css/bootstrap.min.css" var="bootstrapCss" />
<link href="${bootstrapCss}" rel="stylesheet" />
<link href="${coreCss}" rel="stylesheet" />
</head>

<nav class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Project Name</a>
        </div>
    </div>
</nav>

<div class="jumbotron">
    <div class="container">
        <h1>${title}</h1>
        <p>
            <c:if test="${not empty msg}">
                Hello ${msg}
            </c:if>

            <c:if test="${empty msg}">
                Welcome Welcome!
            </c:if>
        </p>
        <p>
            <a class="btn btn-primary btn-lg"
                href="#" role="button">Learn more</a>
        </p>
    </div>
</div>

```

```

        </div>
    </div>

    <div class="container">

        <div class="row">
            <div class="col-md-4">
                <h2>Heading</h2>
                <p>ABC</p>
                <p>
                    <a class="btn btn-default" href="#" role="button">View
details</a>
                </p>
            </div>
            <div class="col-md-4">
                <h2>Heading</h2>
                <p>ABC</p>
                <p>
                    <a class="btn btn-default" href="#" role="button">View
details</a>
                </p>
            </div>
            <div class="col-md-4">
                <h2>Heading</h2>
                <p>ABC</p>
                <p>
                    <a class="btn btn-default" href="#" role="button">View
details</a>
                </p>
            </div>
        </div>

        <hr>
        <footer>
            <p>© Mkyong.com 2015</p>
        </footer>
    </div>

    <spring:url value="/resources/core/css/hello.js" var="coreJs" />
    <spring:url value="/resources/core/css/bootstrap.min.js" var="bootstrapJs" />

    <script src="${coreJs}"></script>
    <script src="${bootstrapJs}"></script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script
>

</body>
</html>

```

3.4 Logging – Send all logs to console.

logback.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>

    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">

            <Pattern>
                %d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n
            </Pattern>

        </layout>
    </appender>

    <logger name="org.springframework" level="debug"
        additivity="false">
        <appender-ref ref="STDOUT" />
    </logger>

    <logger name="com.mkyong.helloworld" level="debug"
        additivity="false">
        <appender-ref ref="STDOUT" />
    </logger>

    <root level="debug">
        <appender-ref ref="STDOUT" />
    </root>

</configuration>

```

4. Spring XML Configuration

Spring XML configuration files.

4.1 Spring root context.

spring-core-config.xml

```

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="com.mkyong.helloworld.service" />

</beans>

```

4.2 Spring web or servlet context.

spring-web-config.xml

```

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd ">

<context:component-scan base-package="com.mkyong.helloworld.web" />

<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="viewClass"
value="org.springframework.web.servlet.view.JstlView"/>
    <property name="prefix" value="/WEB-INF/views/jsp/" />
    <property name="suffix" value=".jsp" />
</bean>

<mvc:resources mapping="/resources/**" location="/resources/" />

<mvc:annotation-driven />

</beans>

```

4.3

Classic web.xml

web.xml

```

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
version="2.5">

<display-name>Gradle + Spring MVC Hello World + XML</display-name>
<description>Spring MVC web application</description>

<!-- For web context -->
<servlet>
    <servlet-name>hello-dispatcher</servlet-name>
    <servlet-class>
        org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring-mvc-config.xml</param-
value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>hello-dispatcher</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>

```

```

<!-- For root context -->
<listener>
    <listener-class>
        org.springframework.web.context.ContextLoaderListener
    </listener-class>
</listener>

<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring-core-config.xml</param-value>
</context-param>

</web-app>

```

5. Demo

- 5.1 The `gradle.build` file is defined an embedded Jetty container. Issues `gradle jettyRun` to start the project.

Terminal

```
your-project$ gradle jettyRun
```

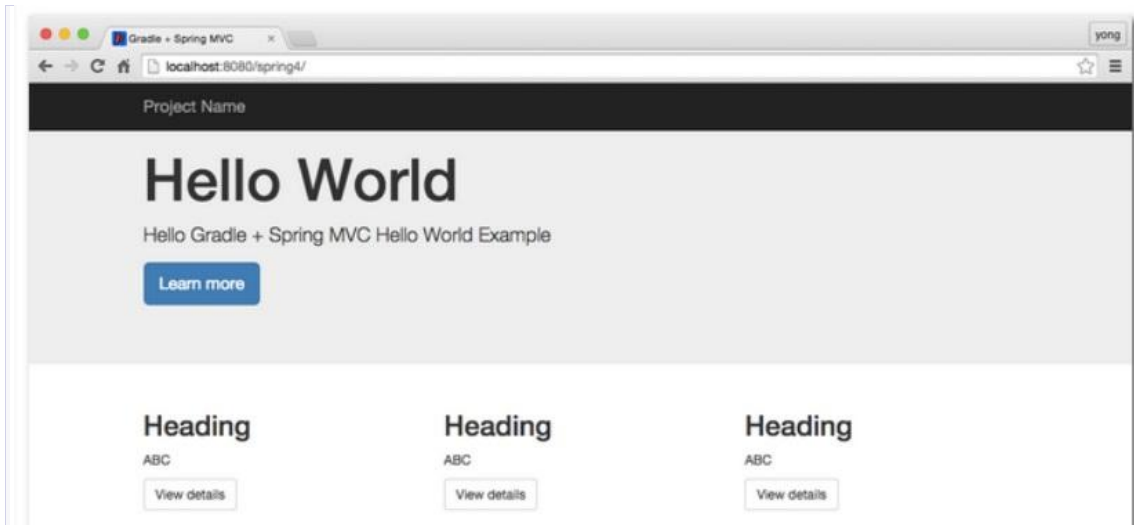
```

:compileJava
:processResources
:classes
:jettyRun
//...SLF4j logging

```

```
> Building 75% > :jettyRun > Running at http://localhost:8080/spring4
```

- 5.2 <http://localhost:8080/spring4/>



6. WAR File

To create a WAR file for deployment :

Terminal

```
your-project$ gradle war
```


A WAR file will be created in **project\build\libs** folder.

\${Project}\build\libs\spring-web-gradle-xml.war