

مثال Hello World در Spring MVC

در مثال زیر چگونگی نوشتن یک برنامه ی ساده ی مبتنی بر وب Hello World با استفاده از Spring MVC Framework نشان داده شده است. برای شروع Eclipse IDE را آماده کنید و جهت توسعه ی یک Dynamic Web Application با استفاده از Spring Web Framework مراحل زیر را دنبال کنید.

مرحله	توضیحات
1	یک Dynamic Web Project با نام HelloWorld ایجاد کنید و در فولدر src موجود در پروژه ی ایجاد شده، بسته ی tutorialspoint.com را ایجاد کنید.
2	Spring زیر و کتابخانه های دیگر را داخل پوشه ی WebContent/WEB-INF/lib.. درگ و دراپ کنید.
3	در بسته ی tutorialspoint.com کلاس جاوای HelloController را ایجاد کنید.
4	در پوشه ی WebContent/WEB-INF پیکربندی اسپرینگ files web.xml و HelloWorld-servlet.xml را ایجاد کنید.
5	در پوشه ی WebContent/WEB-INFfolder زیرپوشه ای به نام jsp ایجاد کنید. در این زیرپوشه یک فایل ویوی hello.jsp ایجاد کنید.
6	مرحله ی آخر ایجاد محتوای فایل های منبع و پیکربندی، همچنین اکسپورت کردن برنامه به صورت زیر است.

HelloController.java

```
package com.tutorialspoint;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.ui.ModelMap;

@Controller
@RequestMapping("/hello")

public class HelloController{

    @RequestMapping(method = RequestMethod.GET)

    public String printHello(ModelMap model) {

        model.addAttribute("message", "Hello Spring MVC Framework!");

        return "hello";

    }

}
```

web.xml

```
<web-app id = "WebApp_ID" version = "2.4"
    xmlns = "http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

    <display-name>Spring MVC Application</display-name>

    <servlet>

        <servlet-name>HelloWeb</servlet-name>

        <servlet-class>
```

```
    org.springframework.web.servlet.DispatcherServlet

    </servlet-class>

    <load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>
    <servlet-name>HelloWeb</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>

</web-app>
```

HelloWeb-servlet.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"
    xsi:schemaLocation = "
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:component-scan base-package = "com.tutorialspoint" />

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

```
</bean>
```

```
</beans>
```

hello.jsp

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

```
<html>
```

```
<head>
```

```
<title>Hello World</title>
```

```
</head>
```

```
<body>
```

```
<h2>${message}</h2>
```

```
</body>
```

```
</html>
```

در ادامه لیست اسپرینگ و کتابخانه های دیگری که قرار است در این برنامه ی وب وارد شوند، ذکر شده اند. می توانیم به راحتی این فایل ها را در پوشه ی WebContent/WEB-INF/lib درگ و دراپ کنیم.

• servlet-api-x.y.z.jar

• commons-logging-x.y.z.jar

• spring-aop-x.y.z.jar

• spring-beans-x.y.z.jar

• spring-context-x.y.z.jar

• spring-core-x.y.z.jar

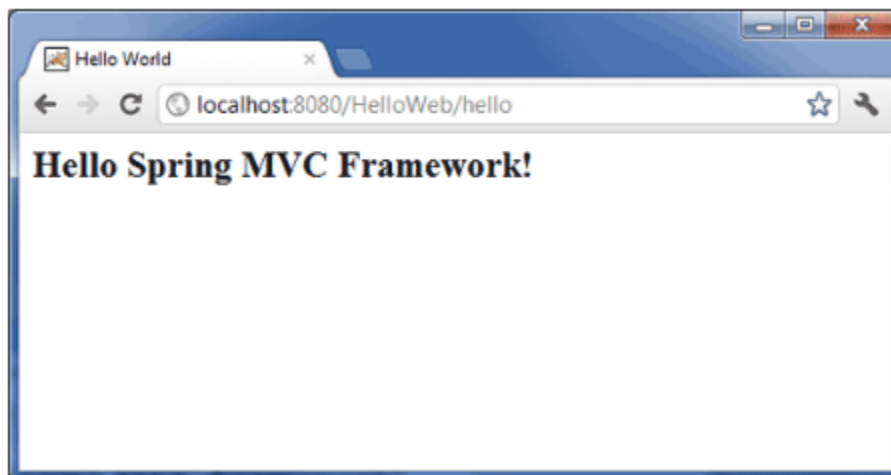
• spring-expression-x.y.z.jar

• spring-webmvc-x.y.z.jar

• spring-web-x.y.z.jar

بعد از این که کار ایجاد فایل های منبع و بیکربندی تمام شد، باید برنامه ی خود را اکسپورت کنید. بر روی برنامه ی خود کلیک راست کنید، از گزینه ی WAR File → Export استفاده کنید و فایل HelloWeb.war خود را در پوشه ی webapps متعلق به Tomcat ذخیره کنید.

حالا سرور Tomcat خود را اجرا کنید و حتما مطمئن شوید که می توانید به صفحات وب دیگر از طریق پوشه ی webapps و با استفاده از یک مرورگر استاندارد دسترسی داشته باشید. حالا سعی کنید به آدرس <http://localhost:8080/HelloWeb/hello> دسترسی پیدا کنید. اگر مشکلی برای برنامه ی Spring Web پیش نیاید، صفحه ی زیر نمایش داده می شود.



توجه داشته باشید که HelloWeb در آدرس موجود اسم برنامه و hello زیرپوشه ی مجازی است که ما با استفاده از `@RequestMapping("/hello")` در کنترلر خود از آن نام برده ایم. می توانید در حین نگاشت آدرس خود با استفاده از `@RequestMapping("/")` از direct root استفاده کنید. در این صورت می توانید با استفاده از آدرس کوتاه <http://localhost:8080/HelloWeb/> به صفحه ی یکسانی دسترسی پیدا کنید. اما توصیه می شود در پوشه های مختلف از قابلیت های مختلف استفاده کنید.