Acabamos de ver como usar Java equals y hashcode

Spring MVC configuration annotation es una de las posibilidades que tenemos en las versiones 3.x y 4.x de Spring Framework a la hora de configurar el framework Spring utilizando unicamente anotaciones y prescindiendo del uso de ficheros XML. Esto permite chequear gran parte de la configuración en tiempo de compilación y evitará muchos problemas.



Spring MVC configuration annotation

Vamos a crear un ejemplo de hola mundo que configure el framework unicamente con anotaciones . Para ello el primer paso es definir un fichero pom.xml con las dependencias del framework que vayamos a utilizar:

```
&amp;lt;dependency&amp;gt;
 &amp;lt;groupId&amp;gt;org.springframework&amp;lt;/groupId&amp;gt;
 &amp;lt;artifactId&amp;gt;spring-webmvc&amp;lt;/artifactId&amp;gt;
 &amp;lt;version&amp;gt;4.2.4.RELEASE&amp;lt;/version&amp;gt;
 &amp;lt;/dependency&amp;gt;
 &amp;lt;dependency&amp;gt;
 &amp;lt;groupId&amp;gt;javax.servlet&amp;lt;/groupId&amp;gt;
 &amp;lt;artifactId&amp;gt;javax.servlet-
api&amp;lt;/artifactId&amp;gt;
 &amp;lt;version&amp;gt;3.1.0&amp;lt;/version&amp;gt;
 &amp;lt;scope&amp;gt;provided&amp;lt;/scope&amp;gt;
 &amp;lt;/dependency&amp;gt;
```

En este caso hemos usado Spring 4 , una vez tenemos las dependencias el siguiente paso es crear una clase de Spring de configuración:

```
package com.arquitecturajava;

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.config.annotation.WebMvcConfigurerAdap
ter;
import
org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration
@EnableWebMvc
@ComponentScan(basePackages = { "com.arquitecturajava" })
public class SpringConfiguracion extends WebMvcConfigurerAdapter{

 @Bean
 public InternalResourceViewResolver getInternalResourceViewResolver()
{
 InternalResourceViewResolver resolver = new
InternalResourceViewResolver();
 resolver.setPrefix("/WEB-INF/jsp/");
 resolver.setSuffix(".jsp");
 return resolver;
 }
```

```
}
```

En esta clase se han definido los packages a escanear a través de la anotación @ComponentScan. Se ha activado las capacidades de Spring MVC utilizando @EnableSpringMVC y finalmente se ha añadido un último bean que referencia al ViewResolver. Vamos a construir un Controlador que mapee la url de /hola y una página jsp dentro de la carpeta /WEB-INF/jsp que nos muestre un mensaje de “hola”.

```
package com.arquitecturajava;

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

@Controller
public class ControladorHola {

@RequestMapping("/hola")
 public String hola() {

 return "hola";
 }

}
```

```
&amp;lt;%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"%&amp;gt;
&amp;lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd"&amp;gt;
&amp;lt;html&amp;gt;
&amp;lt;head&amp;gt;
&amp;lt;meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"&amp;gt;
&amp;lt;title&amp;gt;Insert title here&amp;lt;/title&amp;gt;
&amp;lt;/head&amp;gt;
&amp;lt;body&amp;gt;
hola
&amp;lt;/body&amp;gt;
&amp;lt;/html&amp;gt;
```

Hemos configurado Spring MVC a través del uso de anotaciones , aun así nos queda dar de alta un Servlet que registre Spring MVC como framework en nuestra WebApp.

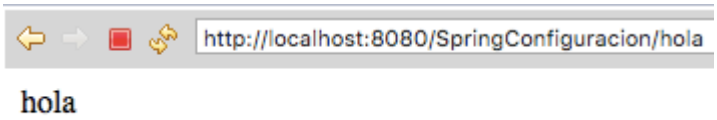
```
&amp;lt;web-app xmlns="http://java.sun.com/xml/ns/javaee"
version="3.1"&amp;gt;

 &amp;lt;servlet&amp;gt;
 &amp;lt;servlet-name&amp;gt;SpringController&amp;lt;/servlet-
name&amp;gt;
 &amp;lt;servlet-
class&amp;gt;org.springframework.web.servlet.DispatcherServlet
 &amp;lt;/servlet-class&amp;gt;
 &amp;lt;init-param&amp;gt;
 &amp;lt;param-name&amp;gt;contextClass&amp;lt;/param-name&amp;gt;
 &amp;lt;param-value&amp;gt;
org.springframework.web.context.support.AnnotationConfigWebApplication
Context
 &amp;lt;/param-value&amp;gt;
```

```
 &amp;lt;/init-param&amp;gt;
 &amp;lt;init-param&amp;gt;
 &amp;lt;param-name&amp;gt;contextConfigLocation&amp;lt;/param-
name&amp;gt;
 &amp;lt;param-value&amp;gt;
 com.arquitecturajava.SpringConfiguracion
 &amp;lt;/param-value&amp;gt;
 &amp;lt;/init-param&amp;gt;
 &amp;lt;/servlet&amp;gt;

 &amp;lt;servlet-mapping&amp;gt;
 &amp;lt;servlet-name&amp;gt;
 SpringController
 &amp;lt;/servlet-name&amp;gt;
 &amp;lt;url-pattern&amp;gt;/&amp;lt;/url-pattern&amp;gt;
 &amp;lt;/servlet-mapping&amp;gt;
 &amp;lt;/web-app&amp;gt;
```

En este caso hemos usado una clase de contexto que se apoya en anotaciones. Invocamos la URL de /hola y la aplicación funcionará sin problemas.



Nos hemos apoyado en Spring MVC configuration annotation para eliminar los ficheros de configuración de Spring Framework. Es posible incluso eliminar el fichero web.xml para ello nos apoyaríamos en un WebApplicationInitializer

Otros artículos relacionados:

Introducción Spring MVC

Spring MVC Controladores