

Blog sobre Java EE

23

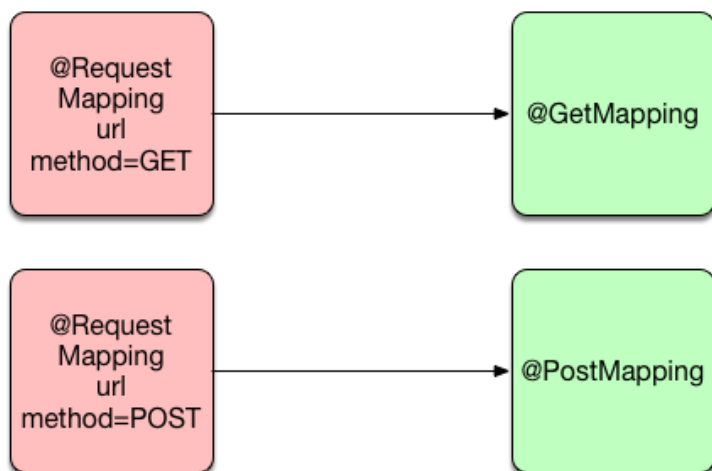
Estás aquí: [Inicio](#)/[Spring](#)/[Spring MVC](#)/Spring GetMapping ,PostMapping etc

23

GetMapping ,PostMapping etc

2017 por [Cecilio Álvarez Caules](#) — 2 comentarios

Spring GetMapping es una anotación de Spring Framework **que aparece con Spring 4.3** y nos permite simplificar el manejo de los diferentes **métodos de Spring MVC** y los **@RequestMapping** que a veces se hacen un poco pesados.



A partir de ahora nosotros podemos usar estas anotaciones **y simplificar el código**. Para ello vamos a construir un ejemplo utilizando **Spring Boot** y con las dependencias de MVC y **Thymeleaf**.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>com.arquitecturajava</groupId>
6   <artifactId>spring</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <packaging>jar</packaging>
```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)
[ACEPTAR](#)

```

19
20     <properties>
21         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
22     23     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEn
23     <java.version>1.8</java.version>
24     </properties>
25
26     <dependencies>
27         <dependency>
28             <groupId>org.springframework.boot</groupId>
29             <artifactId>spring-boot-starter-web</artifactId>
30         </dependency>
31
32     <dependency>
33         <groupId>org.springframework.boot</groupId>
34         <artifactId>spring-boot-starter-thymeleaf</artifactId>
35     </dependency>
36     <dependency>
37         <groupId>org.springframework.boot</groupId>
38         <artifactId>spring-boot-starter-test</artifactId>
39         <scope>test</scope>
40     </dependency>
41 </dependencies>
42
43 <build>
44     <plugins>
45         <plugin>
46             <groupId>org.springframework.boot</groupId>
47             <artifactId>spring-boot-maven-plugin</artifactId>
48         </plugin>
49     </plugins>
50 </build>
51
52
53 </project>

```

El primer paso es ver el código de la aplicación principal:

```

1  package com.arquitecturajava.spring;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5  import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAd
6
7  @SpringBootApplication
8  public class Application extends WebMvcConfigurerAdapter {
9
10     public static void main(String[] args) {
11         SpringApplication.run(Application.class, args);
12     }
13 }

```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

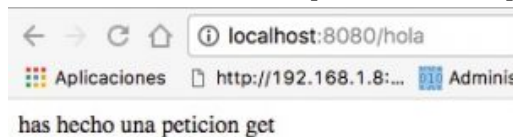
[plugin cookies](#)
[ACEPTAR](#)

```

9  @Controller
10 public class HolaController {
11
12     @GetMapping("/hola")
13     @ResponseBody
14     public String hola() {
15
16         return "has hecho una peticion get";
17     }
18
19     @GetMapping("/formulario")
20     public String hola2() {
21
22         return "formulario";
23     }
24
25     @PostMapping("/holapost")
26     @ResponseBody
27     public String holaPost() {
28
29         return "has hecho una peticion post";
30     }
31
32 }
33

```

Este controlador hace uso de las nuevas anotaciones de **spring** **@GetMapping** ,**@PostMapping** etc. Como se puede ver si solicitamos la url “/hola” nos devolverá “has hecho una peticion get”. Vamos a arrancar la aplicación de SpringBoot y probarlo.



Funciona perfectamente . Una vez que hemos realizado la primera petición solicitamos la carga del formulario con “/formulario”.

```

1  <html>
2  <body>
3
4  <form action="/holapost" method="post">
5  <input type="submit"/>
6  </form>
7
8  </body>
9  </html>

```

COMPARTIR



in

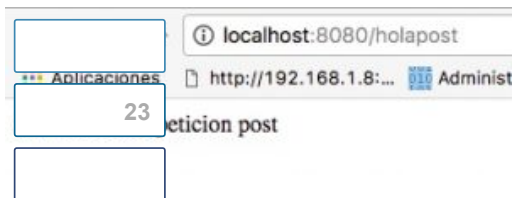
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

Pulsamos el botón y nos mostrará el resultado

23



Acabamos de usar **spring GetMapping y PostMapping** para trabajar con Spring MVC y eliminar el uso de **@RequestMapping**.

Otros artículos relacionados

1. [Spring MVC Flash Attributes](#)
2. [Utilizando Spring MVC Bean Validation](#)
3. [Spring MVC Configuración \(I\)](#)
4. [Spring MVC](#)

Archivada en: [Spring MVC](#)

Comentarios



David dice

28 octubre, 2017 en 18:23

Buenas , como estas ? me parecen interesantes tus articulos, son muy buenos sobre todo por la parte tecnica que le aporta a principiantes. Una cosa que me parece que sumaria un poco mas es que dejes el enlace a un github con el proyecto andando, para poder debuggearlo y poder probarlo en el caso de ver el codigo andando.

[Responder](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

responder

23

 23 comentario

n de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Correo electrónico *

Web

PUBLICAR COMENTARIO

BUSCAR

plugin cookies

arquitecturajava

23

The diagram illustrates the evolution of database architectures through four stages:

- Single Database:** A single database labeled 'dbec' is connected to 'base Data'.
- Sharding:** The data is partitioned into three shards labeled 'A', 'B', and 'C', each with its own 'dbec' instance.
- Replication:** Each shard is replicated across multiple instances. For example, shard 'A' is replicated across three instances, each with its own 'dbec'.
- Distributed System:** The system is fully distributed, with each shard (A, B, C) having its own 'dbec' and 'base Data'.

The diagram also shows a 'mysql' instance connected to the 'base Data'.

The diagram illustrates the architecture of a web application. On the left, a vertical stack of layers is shown within a container labeled 'webapp'. The layers, from top to bottom, are: 'webapp' (the container itself), 'webapp' (a sub-layer), 'db', 'chain', 'webapp', and 'index.html'. On the right, there are three orange circles representing 'SaaS' (Software as a Service) components, grouped by a bracket and labeled 'SaaS'. Below these, there are three orange circles representing 'Services (components)', with an arrow pointing to them from the 'chain' layer. At the bottom right, the text 'arquitecturaxia' is visible.

Spring Boot WAR sin Microservicios

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

23

[Java Herencia vs Interfaces](#)

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

[¿Qué es Spring Boot?](#)[Spring GetMapping ,PostMapping etc](#)[Usando Java Session en aplicaciones web](#)[Java Iterator vs ForEach](#)[Java Constructores this\(\) y super\(\)](#)[Eclipse Git , Repositorios locales y remotos](#)[¿Cuales son las certificaciones Java?](#)[Ejemplo de JPA , Introducción \(I\)](#)[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)[Usando el patron factory](#)[¿Qué es Gradle?](#)[Introducción a Servicios REST](#)[Java Overload y preguntas de certificacion](#)[Uso de Java Generics \(I\)](#)[REST JSON y Java](#)[Mis Libros](#)[Comparando java == vs equals](#)[Spring MVC Configuración \(I\)](#)[Java Override y encapsulación](#)[JPA @GeneratedValue y Primary keys](#)

plugin cookies

ACCEPTAR

23

Copyright © 2017 · eleven40 Pro Theme en Genesis Framework · WordPress · Acceder