




Geeks México


BLOG DE PROGRAMACIÓN EN ESPAÑOL SOBRE JAVA,
FRAMEWORKS, BASES DE DATOS, CÓMPUTO EN LA NUBE, ETC.
EN ESPAÑOL Y EN INGLÉS.

[HOME](#)[ABOUT](#)[CONTACT](#)

Anuncios

**Fine wine or sour milk?**
What's your open source flavor?

SCAN & SEE >



[Report this ad](#)

Primeros pasos con Spring boot 2

[📅 HACE 4 DÍAS](#) [💬 DEJA UN COMENTARIO](#)

Como saben la versión de Spring boot 2 fue liberada hace poco y en Geeks México ya empezamos a hacer pruebas con esta nueva versión.

Paso 1 Configuración

El primer paso será configurar nuestro proyecto, para esto tendremos que configurar los siguientes puntos:

- Proyecto padre: El primer paso será agregar el proyecto padre de Spring boot 2 el cual definirá todas las versiones a utilizar en las starter dependencies.
- Agregar dependencias: Agregaremos la dependencia spring-boot-starter-web para crear un microservicio como ejemplo.
- Configurar plugins: Se definirán 2 plugins, el primero para definir la versión de Java a utilizar (Recordemos que Spring boot 2 solo funciona con Java 8 o superior) y el segundo para generar un fat jar (Una aplicación que contiene un tomcat en si mismo).

[pom.xml](https://github.com/raidentrance/spring-boot2/blob/master/pom.xml) (<https://github.com/raidentrance/spring-boot2/blob/master/pom.xml>)

Con la configuración anterior en el **pom.xml** los 3 puntos mencionados quedarán configurados correctamente.

Paso 2 Creando un Controller

El siguiente paso será crear un controller, en este ejemplo se utilizará Spring mvc:

```
1 | import org.springframework.web.bind.annotation.*;  
2 | import org.springframework.web.bind.annotation.*;  
3 |
```

```
4  /**
5   * @author raidentrance
6   *
7   */
8  @RestController
9  public class SampleController {
10
11      @RequestMapping("/")
12      public String home() {
13          return "Somos Geeks!";
14      }
15  }
```

En el controller anterior devolverá la respuesta Somos Geeks! cuando se ejecute la url <http://localhost:8080/> (<http://localhost:8080/>).

Paso 3 Creando la clase aplicación

Como sabemos Spring boot necesita una clase que contenga un método main para iniciar la aplicación:

```
1  import org.springframework.boot.SpringApplication;
2  import org.springframework.boot.autoconfigure.SpringBootApplication;
3
4  /**
5   * @author raidentrance
6   *
7   */
8  @SpringBootApplication
9  public class SpringBootSampleApplication {
10      public static void main(String[] args) {
11          SpringApplication.run(SpringBootSampleApplication.class, args);
12      }
13  }
```

Al ejecutar la clase SpringBootSampleApplication se ejecutará un servidor tomcat y desplegará nuestra aplicación en el.

Paso 4 Agregar un banner a nuestra aplicación (Opcional)

En la versión anterior de Spring boot el banner era un archivo .txt que se mostraba cuando el servidor iniciaba, en la nueva versión de Spring boot 2 tenemos la posibilidad de poner un gif como banner, para hacerlo

solo debemos poner un archivo con formato .gif, .txt, .jpg o .png con el nombre banner.

En nuestro ejemplo utilizaremos el siguiente gif:



Paso 5 ejecutar nuestra aplicación

El siguiente paso será ejecutar nuestra aplicación, para hacerlo tenemos las siguientes opciones:

- Ejecutar el comando **mvn spring-boot:run**
- Ejecutar la clase que contiene el método **main** **SpringBootApplication**
- Ejecutar el jar generado utilizando **java -jar nombre-de-la-aplicacion.jar**

Al ejecutar cualquiera de las opciones anteriores veremos la siguiente salida:

```

1                                     .##
2                                     &##&oo
3                                     &##oooooooo
4                                     ##oooooooooo
5                                     :##oooooooooo
6                                     .#&oooooooooooo
7                                     #&oooooooooooo
8                                     #o&#.@oooooooo
9                                     #&& @##@oooooooo
10                                    #@ @###@oooooooo
11                                    #@:###:@oooooooo
12                                    #&###@ @oooooooo
13                                    &.####@@oooooooo
14                                    ##@#####@oooooooo
15                                    #ooo&@#@oooooooo

```

```

16      :&#:ooooooooooooo
17      *##o#oooooooooooo
18      .#o8oo#&oooooooo
19      ##o#&oo###:c
20      :#&oo#&ooo
21      .##&o8#
22      :ooooo: #####8oooooooooooo:##oo##
23      ooooooooo***:ooooooooooooooooo#oooooooo
24      .**oooooooo:***:o&#####8&o##&oooooooo
25      ****oooo*****ooo      ##oooooooooooo
26      ***oooo*****:oo      :oooooooooooo
27      **:o***** <img draggable="false" cla
28      *****...:      ###&&&ooooo
29      ***** *      &#oooooooooooo
30      ***:oo      #&oooooooooooo
31      *.      #oooooooooooo
32      #&oooooooooooo
33      ##oooooooooooo
34      :#oooooooooooo
35      ##oooooooooooo
36      ##oo:oooo
37      ##o#o###f
38      ##&oooooooo
39      ##&&oooo
40
41
42
43 2018-03-07 15:36:40.630 INFO 43046 --- [
44 2018-03-07 15:36:40.632 INFO 43046 --- [
45 2018-03-07 15:36:40.664 INFO 43046 --- [
46 2018-03-07 15:36:41.381 INFO 43046 --- [
47 2018-03-07 15:36:41.403 INFO 43046 --- [
48 2018-03-07 15:36:41.403 INFO 43046 --- [
49 2018-03-07 15:36:41.413 INFO 43046 --- [c
50 2018-03-07 15:36:41.481 INFO 43046 --- [c
51 2018-03-07 15:36:41.481 INFO 43046 --- [c
52 2018-03-07 15:36:41.587 INFO 43046 --- [c
53 2018-03-07 15:36:41.590 INFO 43046 --- [c
54 2018-03-07 15:36:41.590 INFO 43046 --- [c
55 2018-03-07 15:36:41.591 INFO 43046 --- [c
56 2018-03-07 15:36:41.591 INFO 43046 --- [c
57 2018-03-07 15:36:41.833 INFO 43046 --- [
58 2018-03-07 15:36:41.892 INFO 43046 --- [
59 2018-03-07 15:36:41.897 INFO 43046 --- [
60 2018-03-07 15:36:41.898 INFO 43046 --- [
61 2018-03-07 15:36:41.926 INFO 43046 --- [
62 2018-03-07 15:36:41.926 INFO 43046 --- [
63 2018-03-07 15:36:41.954 INFO 43046 --- [
64 2018-03-07 15:36:42.055 INFO 43046 --- [
65 2018-03-07 15:36:42.096 INFO 43046 --- [
66 2018-03-07 15:36:42.100 INFO 43046 --- [

```

Una vez que el servidor se inició correctamente podemos ejecutar nuestro endpoint accediendo a la url <http://localhost:8080/> (<http://localhost:8080/>) , mostrando la siguiente salida:

Puedes encontrar el código completo en el siguiente enlace <https://github.com/raidentrance/spring-boot2/tree/master/src/main> (<https://github.com/raidentrance/spring-boot2/tree/master/src/main>).


Si te gusta el contenido y quieres enterarte cuando realicemos un post nuevo síguenos en nuestras redes sociales https://twitter.com/geeks_mx (https://twitter.com/geeks_mx) y <https://www.facebook.com/geeksJavaMexico/> (<https://www.facebook.com/geeksJavaMexico/>).

Autor: Alejandro Agapito Bautista

Twitter: @raidentrance

Contacto:raidentrance@gmail.com

Anuncios




**Test Automation with Katalon Studio
(Selenium WebDriver)**

Katalon Studio - Keyword driven tool written on top of
Selenium for Web Application Automation

Shop now

[Report this ad](#)



**Test Automation with Katalon Studio
(Selenium WebDriver)**

Katalon Studio - Keyword driven tool written on top of
Selenium for Web Application Automation

Shop now

[Report this ad](#)

