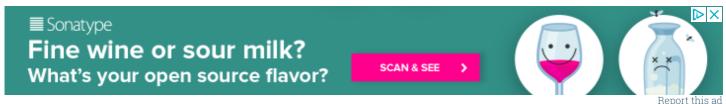


Geeks México

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



Primeros pasos con Spring boot 2

 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)

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:

```
import org.springframework.web.bind.annota
import org.springframework.web.bind.annota
```

```
4
5
        @author raidentrance
6
 7
8
     @RestController
9
     public class SampleController {
10
         @RequestMapping("/")
11
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.SpringApp]
2
     import org.springframework.boot.autoconfig
3
4
5
        @author raidentrance
6
7
8
     @SpringBootApplication
9
     public class SpringBootSampleApplication {
10
         public static void main(String[] args)
11
             SpringApplication.run(SpringBootSa
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 **SpringBootSampleApplication**
- Ejecutar el jar generado utilizando java -jar nombre-de-la-aplicacion.jar

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

.###	1
&##&ood</td><td>2</td></tr><tr><td>&##000000</td><td>3</td></tr><tr><td>##00000000</td><td>4</td></tr><tr><td>:##000000000</td><td>5</td></tr><tr><td>.#&0000000000</td><td>6</td></tr><tr><td>#&0000000000</td><td>7</td></tr><tr><td>#0&#.@0000000</td><td>8</td></tr><tr><td>#&& @#@000000</td><td>9</td></tr><tr><td>#@ @###@000000</td><td>10</td></tr><tr><td>#@:###:@000000</td><td>11</td></tr><tr><td>#&###@ @oooooo</td><td>12</td></tr><tr><td>&.###<u>@</u>@ooooo</td><td>13</td></tr><tr><td>##@@###@ooooo</td><td>14</td></tr><tr><td>#000&@@#0000000</td><td>15</td></tr><tr><td>==</td><td></td></tr></tbody></table>	

```
16
                                :&#:00000000000
17
                                 *##0#000000000
18
                                   .#0800#&00000
19
                                    ##0#&00###:0
20
                                       :#&00#&000
21
                                          .##&08#
22
                 :00000:
                          ####800000000:##00###
23
             0000000***:00000000000000#0000000
24
           **ooooo:***:o&#######8&o##&oooooc
25
         ****0000*****
                                     ##00000000
         ***0000****:00
26
                                     :0000000000
          **:o**** <img draggable="false" clas
27
        *****
                                     ###&&ooooc
28
        *****
29
                                    &#000000000C
       ***:00
                                    #&000000000
30
31
                                    #0000000000
32
                                    #&000000000
33
                                    ##000000000
34
                                     :#000000000
35
                                      ##00000000
36
                                       ##00:0000
37
                                       ##0#0###
38
                                       ##&000000
39
                                        ##&&oooc
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
                               INFO 43046 ---
     2018-03-07 15:36:41.413
50
     2018-03-07 15:36:41.481
                               INFO 43046 ---
51
     2018-03-07 15:36:41.481
                               INFO 43046 ---
52
     2018-03-07 15:36:41.587
                               INFO 43046 ---
53
     2018-03-07 15:36:41.590
                               INFO 43046 ---
                                               Γc
54
     2018-03-07 15:36:41.590
                               INFO 43046 ---
55
     2018-03-07 15:36:41.591
                               INFO 43046 ---
                                               Γ
     2018-03-07 15:36:41.591
56
                               INFO 43046 ---
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 ---
     2018-03-07 15:36:41.898
60
                               INFO 43046 ---
     2018-03-07 15:36:41.926
                               INFO 43046 ---
61
62
     2018-03-07 15:36:41.926
                               INFO 43046 ---
     2018-03-07 15:36:41.954
                               INFO 43046 ---
63
     2018-03-07 15:36:42.055
                               INFO 43046 ---
64
65
     2018-03-07 15:36:42.096
                               INFO 43046 ---
     2018-03-07 15:36:42.100
                               INFO 43046 ---
66
```

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:

Somos Geeks!

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).

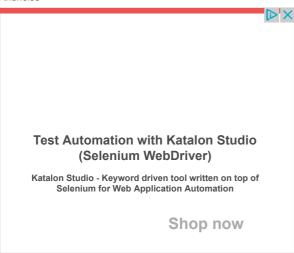
Si te gusta el contenido y quieres enterarte cuando realicemos un post nuevo síguenos en nuestras redes sociales https://twitter.com/geeks_mx (https://twitter.com/geeks_mx) y https://www.facebook.com/geeksJavaMexico/ (https://www.facebook.com/geeksJavaMexico/).

Autor: Alejandro Agapito Bautista

Twitter: @raidentrance

Contacto:raidentrance@gmail.com

Anuncios



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