

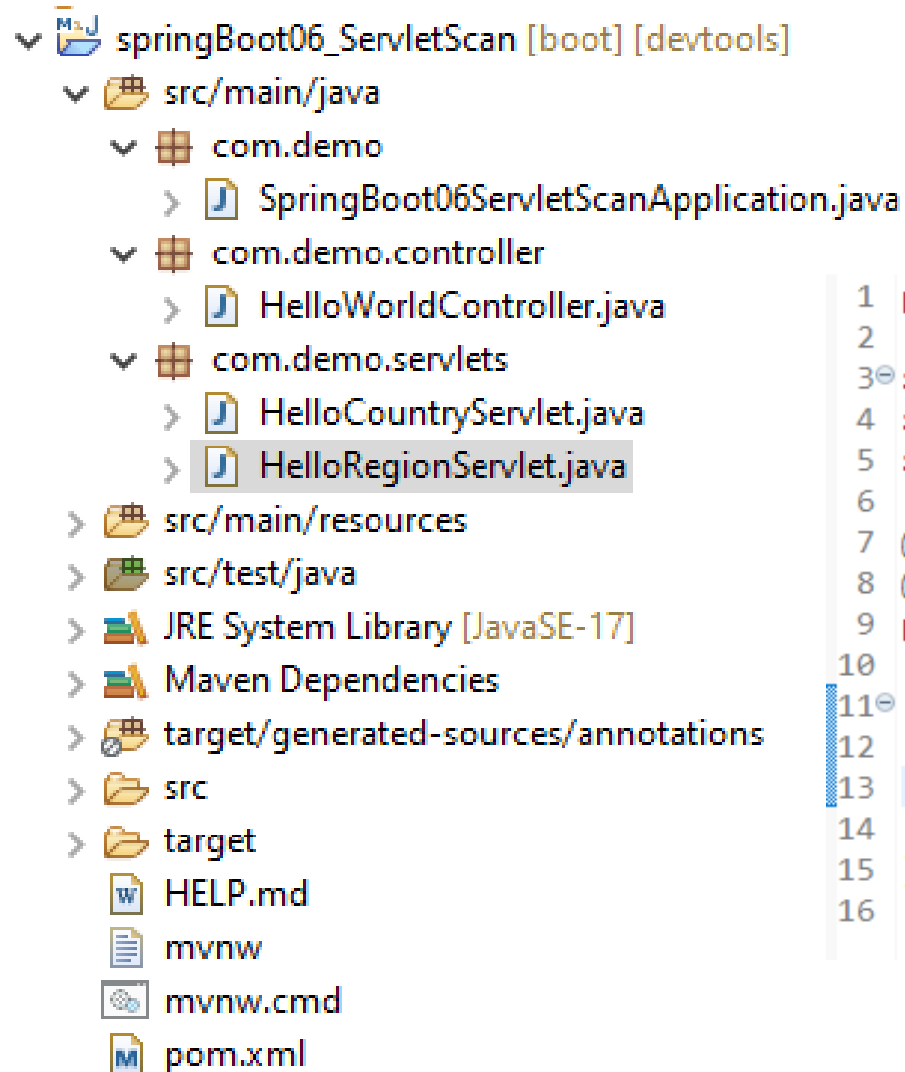
Spring Boot Servlet Mapping

Springboot06, springboot07

Spring Boot Servlet

- ❑ La radice di Spring Boot Web e Spring MVC è la **DispatcherServlet**
- ❑ Tutte le **richieste HTTP** vengono intercettate e smistate verso le view opportune
- ❑ Spring Boot fornisce il supporto per utilizzare servlet in modo **nativo**, gestendo le chiamate HTTP direttamente, come in qualsiasi applicazione Java EE
- ❑ Con Spring Boot si possono gestire le servlet in due modi:
 - Marcando la classe che estende `HttpServlet` con ***@WebServlet*** e con ***@ServletComponentScan*** allo startup dell'applicazione
 - Registrando la classe che estende `HttpServlet` come un **bean** Spring usando **`ServletRegistrationBean`**

Spring Boot Servlet Component Scan



```
1 package com.demo;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.boot.web.servlet.ServletComponentScan;
6
7 @SpringBootApplication
8 @ServletComponentScan
9 public class SpringBoot06ServletScanApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(SpringBoot06ServletScanApplication.class, args);
13     }
14
15 }
16
```

Spring Boot Servlet Component Scan

- ❑ **@ServletComponentScan** effettua scan di Servlets annotate con **@WebServlet**
Filtri annotati con **@WebFilter**, Listener annotati con **@WebListener**
- ❑ **@ServletComponentScan** cerca Servlets, Filters e Listeners solo usando embedded web servers

```
SpringBoot06ServletScanApplication.java ×  
1 package com.demo;  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5 import org.springframework.boot.web.servlet.ServletComponentScan;  
6  
7 @SpringBootApplication  
8 @ServletComponentScan  
9 public class SpringBoot06ServletScanApplication {  
10  
11     public static void main(String[] args) {  
12         SpringApplication.run(SpringBoot06ServletScanApplication.class, args);  
13     }  
14  
15 }
```

Spring Boot Servlet Component Scan

```
SpringBoot06ServletScanApplication.java HelloCountryServlet.java X
1 package com.demo.servlets;
2 import java.io.IOException;
3 import java.io.PrintWriter;
4 import jakarta.servlet.annotation.WebServlet;
5 import jakarta.servlet.http.HttpServlet;
6 import jakarta.servlet.http.HttpServletRequest;
7 import jakarta.servlet.http.HttpServletResponse;
8
9 @WebServlet(urlPatterns = "/country/*", loadOnStartup = 1)
10 public class HelloCountryServlet extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12     public void doPost(HttpServletRequest request, HttpServletResponse response)
13         throws IOException{
14         doGet(request,response);
15     }
16     public void doGet(HttpServletRequest request, HttpServletResponse response)
17         throws IOException {
18         response.setContentType("text/html");
19         PrintWriter out = response.getWriter();
20         out.println("<h3>Hello Italy!</h3>");
21     }
22
23 }
```

Spring Boot Servlet Component Scan

```
SpringBoot06ServletScanApplication.java  HelloCountryServlet.java  HelloRegionServlet.java  X
1  package com.demo.servlets;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5
6  import jakarta.servlet.annotation.WebServlet;
7  import jakarta.servlet.http.HttpServlet;
8  import jakarta.servlet.http.HttpServletRequest;
9  import jakarta.servlet.http.HttpServletResponse;
10
11
12  @WebServlet(urlPatterns = "/region/*", loadOnStartup = 1)
13  public class HelloRegionServlet extends HttpServlet {
14      private static final long serialVersionUID = 1L;
15      public void doPost(HttpServletRequest request, HttpServletResponse response)
16                          throws IOException{
17          doGet(request,response);
18      }
19      public void doGet(HttpServletRequest request, HttpServletResponse response)
20                          throws IOException {
21          response.setContentType("text/html");
22          PrintWriter out = response.getWriter();
23          out.println("<h3>Hello Piemonte!</h3>");
24      }
25  }
```


24/04/2023

Spring Boot Servlet Registration Bean

▼  springBoot07_ServletBean [boot] [devtools]

▼  src/main/java

▼  com.demo

>  SpringBootApplication.java

>  WebConfig.java

▼  com.demo.controller

>  HelloWorldController.java

▼  com.demo.servlets

>  HelloCountryServlet.java

>  HelloRegionServlet.java

>  src/main/resources

>  JRE System Library [JavaSE-17]

>  Maven Dependencies

>  src

>  target

>  pom.xml

```
1 package com.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class SpringBootApplication {
8     public static void main(String[] args) {
9         SpringApplication.run(SpringBootApplication.class, args);
10    }
11 }
12
```

Spring Boot Servlet Registration Bean

- ❑ Le Servlet vengono registrate da ***ServletRegistrationBean*** ed è necessario creare un bean di questo tipo in configurazione
- ❑ Alcuni metodi usati per configurare le Servlet sono:
 - **setServlet()**: imposta la Servlet da registrare
 - **addUrlMappings()**: Aggiunge il mapping dell'URL per la Servlet
 - **setLoadOnStartup**: Imposta la priorità di caricamento allo startup
- ❑ Questa modalità, in luogo dell'utilizzo delle annotation ***@WebServlet*** e ***@ServletComponentScan*** non sfrutta le capacità di autoconfigurazione ma permette un controllo più completo sul mapping delle servlet

Spring Boot Servlet Registration Bean

```
SpringBoot0... HelloCountry... HelloWorldC... × SpringBootA... WebConfig.java × HelloCountry... HelloRegi...
1 package com.demo;
2
3 import org.springframework.boot.web.servlet.ServletRegistrationBean;
4 import org.springframework.context.annotation.Bean;
5 import org.springframework.context.annotation.Configuration;
6
7 import com.demo.servlets.HelloCountryServlet;
8 import com.demo.servlets.HelloRegionServlet;
9
10 import jakarta.servlet.http.HttpServlet;
11
12 @Configuration
13 public class WebConfig {
14     @Bean
15     public ServletRegistrationBean<HttpServlet> countryServlet() {
16         ServletRegistrationBean<HttpServlet> servRegBean = new ServletRegistrationBean<>();
17         servRegBean.setServlet(new HelloCountryServlet());
18         servRegBean.addUrlMappings("/country/*");
19         servRegBean.setLoadOnStartup(1);
20         return servRegBean;
21     }
22     @Bean
23     public ServletRegistrationBean<HttpServlet> stateServlet() {
24         ServletRegistrationBean<HttpServlet> servRegBean = new ServletRegistrationBean<>();
25         servRegBean.setServlet(new HelloRegionServlet());
26         servRegBean.addUrlMappings("/region/*");
27         servRegBean.setLoadOnStartup(1);
28         return servRegBean;
29     }
30 }
```

Spring Boot Servlet Registration Bean



The screenshot shows an IDE with several tabs: SpringBoot0..., HelloWorldC..., WebConfig.java, HelloCountry..., and HelloRegion. The active tab is WebConfig.java, which contains the following Java code:

```
1 package com.demo.controller;
2 import org.springframework.web.bind.annotation.RequestMapping;
3 import org.springframework.web.bind.annotation.RestController;
4
5 @RestController
6 public class HelloWorldController {
7     volatile int counter=0;
8     @RequestMapping("/world")
9     public String helloMsg() {
10         String msg = "Hello World! " + counter++;
11         return msg;
12     }
13 }
14
```

Below the code editor, there is a toolbar with icons for Console, Problems, Progress, Debug Shell, and Internal Web Browser. The Internal Web Browser tab is active, showing the URL <http://localhost:8080/world> and the response `Hello World! 6`.

Spring Boot Servlet Registration Bean



The screenshot shows an IDE with several tabs open: SpringBoot0..., HelloWorldC..., WebConfig.java, HelloCountry..., HelloRegion..., url.txt, and another url.txt. The main editor displays the following Java code:

```
1 package com.demo.servlets;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import jakarta.servlet.http.HttpServlet;
7 import jakarta.servlet.http.HttpServletRequest;
8 import jakarta.servlet.http.HttpServletResponse;
9
10 public class HelloCountryServlet extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12     volatile int counter=0;
13     public void doPost(HttpServletRequest request, HttpServletResponse response)
14         throws IOException{
15         doGet(request,response);
16     }
17     public void doGet(HttpServletRequest request, HttpServletResponse response)
18         throws IOException {
19         response.setContentType("text/html");
20         PrintWriter out = response.getWriter();
21         out.println("<h3>Hello Italy! "+counter+ "</h3>");
22         counter++;
23     }
24 }
```

To the right, a web browser window is open at the URL `http://localhost:8080/country`, displaying the output: **Hello Italy! 3**.

Spring Boot Servlet Registration Bean

```
SpringBoot0... HelloWorldC... WebConfig.java HelloCountry... HelloRegion... url.txt »5
1 package com.demo.servlets;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import jakarta.servlet.http.HttpServlet;
7 import jakarta.servlet.http.HttpServletRequest;
8 import jakarta.servlet.http.HttpServletResponse;
9
10
11 public class HelloRegionServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13     volatile int counter=0;
14     public void doPost(HttpServletRequest request, HttpServletResponse response)
15         throws IOException{
16         doGet(request,response);
17     }
18     public void doGet(HttpServletRequest request, HttpServletResponse response)
19         throws IOException {
20         response.setContentType("text/html");
21         PrintWriter out = response.getWriter();
22         out.println("<h3>Hello Piemonte! " + counter + "</h3>");
23         counter++;
24     }
25 }
```

