Spring Boot DevTools es la herramienta de Spring Boot que nos permite reiniciar de forma automática nuestras aplicaciones cada vez que se produce un cambio en nuestro código. Vamos a ver un ejemplo sencillo. Para ello partiremos del típico código con un servicio REST de Hola que nos devuelve un mensaje.

```
package com.arquitecturajava.web.servicios;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HolaService {

	@GetMapping("/hola")
	public String hola() {
		return "hola";
	}
}
```

Este servicio cuando le invocamos nos devolverá a nivel de datos el mensaje de hola.



El problema que tenemos con Spring Boot es que si cambiamos el código fuente el servicio no se actualizará tendremos que parar el servidor y volverlo a arrancar para que el nuevo código funcione. Esto a la larga se convierte en un problema.

```
package com.arquitecturajava.web.servicios;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HolaService {

	@GetMapping("/hola")
	public String hola() {
		return "hola 2";
	}
}
```

Spring Boot DevTools

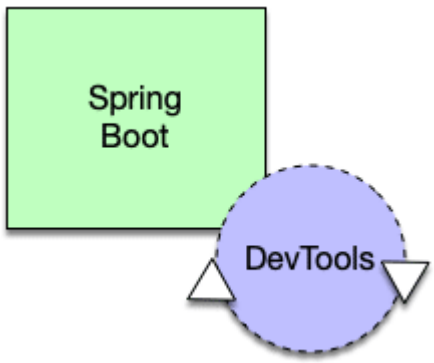
Para solventar este problema disponemos de SpringBoot DevTools estas herramientas se instalan como Spring Boot Starter y nos permiten que cada vez que realicemos un cambio en el servidor la aplicación se actualize y reinicie sola de forma totalmente transparente .

```
<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-
web</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-devtools</artifactId>
		</dependency>

		<dependency>
```

```
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-
test</artifactId>
                        <scope>test</scope>
                        <exclusions>
                                <exclusion>
<groupId>org.junit.vintage</groupId>
                                        <artifactId>junit-vintage-
engine</artifactId>
                                </exclusion>
                        </exclusions>
                </dependency>
        </dependencies>
```

De esta manera cada vez que realicemos un cambio en la aplicación esta se reinicia de forma automática.



El resultado se muestra en pantalla y la aplicación se recarga de forma automática una vez tenemos la herramienta configurada.



Otros artículos relacionados

- ¿Que es Spring Boot?
- ¿Que es Spring WebFlux?
- Spring Boot JPA
- Spring Initializer