


nuncios



```
<?php find_developers( [ 'language' => 'PHP',  
  'specialty' => 'scaling', 'location' => ANYWHERE ] )
```

[APPLY](#)

REPORT THIS AD

**devs4j**

EL MEJOR SITIO PARA SABER SOBRE PROGRAMACIÓN EN ESPAÑOL.

ESPAÑOL

Spring framework 5 : Uso de los scopes (prototype y singleton) utilizando la anotación @Scope

BY RAIDENTRANCE ON FEBRERO 7, 2019 · ([DEJA UN COMENTARIO](#))

1 Vote

El scope de un bean define su ciclo de vida, en los posts anteriores hemos creado beans pero no hemos definido su scope, al hacer esto Spring le asigna uno por defecto, a continuación se muestran los scopes disponibles:

Scope	Descripción
Singleton	Crea una sola instancia del bean por contenedor de Spring
Prototype	Crea una nueva instancia cada vez que se solicita
Request	Crea una nueva instancia por cada petición HTTP, solo se puede utilizar en una aplicación web
Session	Crea una nueva instancia por cada sesión HTTP
Application	Crea una nueva instancia por cada ServletContext

En este post nos enfocaremos en los scopes Singleton y Prototype.

Scope singleton

El scope singleton es el scope por defecto en spring, veamos el siguiente bean:

```
import org.springframework.stereotype.Component;

/**
 * @author raidentrance
 *
 */
@Component
public class SingletonScopeBean {
}
```

Ahora veamos la siguiente clase aplicación:

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ConfigurableApplicationContext;

import com.devs4j.scopes.SingletonScopeBean;

@SpringBootApplication
public class SpringExampleApplication {

    public static void main(String[] args) {
        ConfigurableApplicationContext appli

        SingletonScopeBean singletonBean = a

        SingletonScopeBean secondSingletonBe

        System.out.println(singletonBean.equ

    }

}
```

La salida será la siguiente:

```
true
```

Podemos ver que tenemos 2 beans **singletonBean** y **secondSingletonBean** y al llamar el método equals la salida es true, indicando que es el mismo objeto.

Scope prototype

El scope prototype indica que se creará un bean nuevo cada que se mande llamar, por esto re utilizaremos el bean anterior y solo cambiaremos su scope como se muestra a continuación:

```
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("prototype")
public class PrototypeScopeBean {

}
```

Re utilizaremos la clase aplicación:

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ConfigurableApplicationContext;

import com.devs4j.scopes.PrototypeScopeBean;

@SpringBootApplication
public class SpringExampleApplication {

    public static void main(String[] args) {
        ConfigurableApplicationContext appli

        PrototypeScopeBean prototypeBean = a

        PrototypeScopeBean secondPrototypeBe
```

```
        System.out.println (prototypeBean.equals(prototypeBean));  
    }  
}
```

La salida será la siguiente:

```
false
```

Esto se debe a que cada vez que se mande llamar el método `getBean` se creará un bean nuevo, es por esto que al mandar llamar al método `equals` el resultado será **false**.

Para estar al pendiente sobre nuestro contenido nuevo síguenos en nuestras redes sociales <https://www.facebook.com/devs4j/>

(<https://www.facebook.com/devs4j/>) y <https://twitter.com/devs4j> (<https://twitter.com/devs4j>).

Autor: Alejandro Agapito Bautista

Twitter: @raidentrance

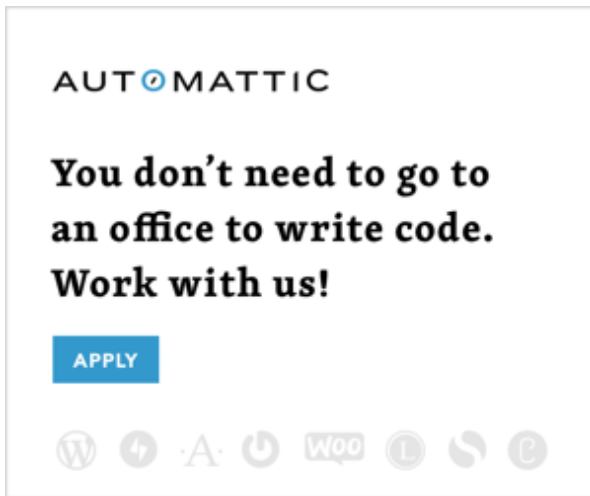
(<https://geeksjavamexico.wordpress.com/mentions/raidentrance/>)

Contacto:raidentrance@gmail.com

Anuncios



REPORT THIS AD



REPORT THIS AD



Publicado por raidentrance

Soy @raidentrance en Twitter y en Github, soy egresado de la Facultad de Ingeniería de la UNAM, cuento con 8 certificaciones en diferentes áreas del desarrollo de software, me gustan las cervezas y soy Geek. Ver todas las entradas de raidentrance (<https://devs4j.com/author/raidentrance/>)

