

(<http://baeldung.com>)

Una guía rápida de Spring @Lazy Annotation

Última modificación: 23 de marzo de 2018

por [baeldung](http://www.baeldung.com/author/baeldung/) (<http://www.baeldung.com/author/baeldung/>)
(<http://www.baeldung.com/author/baeldung/>)

Primavera (<http://www.baeldung.com/category/spring/>) +

Acabo de anunciar los nuevos módulos de *Spring 5* en REST With Spring:

>> **COMPRUEBA EL CURSO** (</rest-with-spring-course#new-modules>)

1. Información general

De forma predeterminada, Spring crea todos los beans simples con entusiasmo en el arranque / arranque del contexto de la aplicación. La razón detrás de esto es simple: para evitar y detectar todos los posibles errores de inmediato en lugar

de en tiempo de ejecución.

Sin embargo, hay casos en los que necesitamos crear un bean, no en el inicio del contexto de la aplicación, sino cuando lo solicitamos.

En este tutorial rápido, vamos a analizar la anotación `@Lazy` de Spring .

2. Inicialización lenta

La anotación `@Lazy` ha estado presente desde Spring versión 3.0. Hay varias maneras de decirle al contenedor de IoC que inicialice un bean perezosamente.

2.1. `@Configuration` Class

Cuando **colocamos la anotación `@Lazy` sobre la clase `@Configuration`**, indica que todos los métodos con la anotación `@Bean` deben cargarse de forma diferida .

Este es el equivalente para el atributo `default-lazy-init = "true "` de la configuración basada en XML .

Echemos un vistazo aquí:

```
1  @Lazy
2  @Configuration
3  @ComponentScan(basePackages = "com.baeldung.lazy")
4  public class AppConfig {
5
6      @Bean
7      public Region getRegion(){
8          return new Region();
9      }
10
11     @Bean
12     public Country getCountry(){
13         return new Country();
14     }
15 }
```

Probemos ahora la funcionalidad:

```
1  @Test
2  public void givenLazyAnnotation_whenConfigClass_thenLazyAll() {
3
4      AnnotationConfigApplicationContext ctx
5          = new AnnotationConfigApplicationContext();
6      ctx.register(AppConfig.class);
7      ctx.refresh();
8      ctx.getBean(Region.class);
9      ctx.getBean(Country.class);
10 }
```

Como vemos, todos los beans se crean solo cuando los solicitamos por primera vez:

```
1  Bean factory for ...AnnotationConfigApplicationContext:
2  ...DefaultListableBeanFactory: [...];
3  // application context started
4  Region bean initialized
5  Country bean initialized
```

Para aplicar esto solo a un bean específico, eliminemos el *@Lazy* de una clase.

Luego lo agregamos a la configuración del bean deseado:

```
1  @Bean
2  @Lazy(true)
3  public Region getRegion(){
4      return new Region();
5  }
```

2.2 con *@Autowired*

Antes de seguir adelante, consulte estas guías para las anotaciones *@Autowired* (<http://www.baeldung.com/spring-autowire>) y *@Component* (<http://www.baeldung.com/spring-bean-annotations>).

Aquí, **para inicializar un bean perezoso, lo referenciamos desde otro.**

El frijol que queremos cargar perezosamente:

```
1  @Lazy
2  @Component
3  public class City {
4      public City() {
5          System.out.println("City bean initialized");
6      }
7  }
```

Y es referencia:

```
1 public class Region {
2
3     @Lazy
4     @Autowired
5     private City city;
6
7     public Region() {
8         System.out.println("Region bean initialized");
9     }
10
11     public City getCityInstance() {
12         return city;
13     }
14 }
```

Tenga en cuenta que *@Lazy* es obligatorio en ambos lugares.

Con la anotación *@Component* en la clase *City* y al hacer referencia a ella con *@Autowired*:

```
1 @Test
2 public void givenLazyAnnotation_whenAutowire_thenLazyBean() {
3     // load up ctx application context
4     Region region = ctx.getBean(Region.class);
5     region.getCityInstance();
6 }
```

Aquí, **el bean *City* se inicializa solo cuando llamamos al método *getCityInstance()*.**

3. Conclusión

En este rápido tutorial, aprendimos los conceptos básicos de la anotación *@Lazy* de Spring . Examinamos varias formas de configurarlo y usarlo.

Como de costumbre, el código completo para esta guía está disponible en GitHub (<https://github.com/eugenp/tutorials/tree/master/spring-core>) .

Acabo de anunciar los nuevos módulos de Spring 5 en REST With Spring:

[>> VERIFIQUE LAS LECCIONES \(/rest-with-spring-course#new-modules\)](#)

Deja una respuesta

¡Sé el primero en comentar!



Start the discussion

✉ Suscribirse ▼

CATEGORÍAS

PRIMAVERA ([HTTP://WWW.BAELDUNG.COM/CATEGORY/SPRING/](http://www.baeldung.com/category/spring/))

DESCANSO ([HTTP://WWW.BAELDUNG.COM/CATEGORY/REST/](http://www.baeldung.com/category/rest/))

JAVA ([HTTP://WWW.BAELDUNG.COM/CATEGORY/JAVA/](http://www.baeldung.com/category/java/))

SEGURIDAD ([HTTP://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/](http://www.baeldung.com/category/security-2/))

PERSISTENCIA ([HTTP://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/](http://www.baeldung.com/category/persistence/))

JACKSON ([HTTP://WWW.BAELDUNG.COM/CATEGORY/JACKSON/](http://www.baeldung.com/category/jackson/))

HTTPCLIENT ([HTTP://WWW.BAELDUNG.COM/CATEGORY/HTTP/](http://www.baeldung.com/category/http/))

KOTLIN ([HTTP://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/](http://www.baeldung.com/category/kotlin/))

SERIE

TUTORIAL "VOLVER A LO BÁSICO" DE JAVA ([HTTP://WWW.BAELDUNG.COM/JAVA-TUTORIAL](http://www.baeldung.com/java-tutorial))

JACKSON JSON TUTORIAL ([HTTP://WWW.BAELDUNG.COM/JACKSON](http://www.baeldung.com/jackson))

TUTORIAL DE HTTPCLIENT 4 ([HTTP://WWW.BAELDUNG.COM/HTTPCLIENT-GUIDE](http://www.baeldung.com/httpclient-guide))

REST CON SPRING TUTORIAL ([HTTP://WWW.BAELDUNG.COM/REST-WITH-SPRING-SERIES/](http://www.baeldung.com/rest-with-spring-series/))

TUTORIAL DE SPRING PERSISTENCE ([HTTP://WWW.BAELDUNG.COM/PERSISTENCE-WITH-SPRING-SERIES/](http://www.baeldung.com/persistence-with-spring-series/))

SEGURIDAD CON SPRING ([HTTP://WWW.BAELDUNG.COM/SECURITY-SPRING](http://www.baeldung.com/security-spring))

ACERCA DE

ACERCA DE BAELDUNG ([HTTP://WWW.BAELDUNG.COM/ABOUT/](http://www.baeldung.com/about/))

LOS CURSOS ([HTTP://COURSES.BAELDUNG.COM](http://courses.baeldung.com))

TRABAJO DE CONSULTORÍA ([HTTP://WWW.BAELDUNG.COM/CONSULTING](http://www.baeldung.com/consulting))

META BAELDUNG ([HTTP://META.BAELDUNG.COM/](http://meta.baeldung.com/))

EL ARCHIVO COMPLETO ([HTTP://WWW.BAELDUNG.COM/FULL_ARCHIVE](http://www.baeldung.com/full_archive))

ESCRIBIR PARA BAELDUNG ([HTTP://WWW.BAELDUNG.COM/CONTRIBUTION-GUIDELINES](http://www.baeldung.com/contribution-guidelines))

CONTACTO ([HTTP://WWW.BAELDUNG.COM/CONTACT](http://www.baeldung.com/contact))

INFORMACIÓN DE LA COMPAÑÍA ([HTTP://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO](http://www.baeldung.com/baeldung-company-info))

TÉRMINOS DE SERVICIO ([HTTP://WWW.BAELDUNG.COM/TERMS-OF-SERVICE](http://www.baeldung.com/terms-of-service))

POLÍTICA DE PRIVACIDAD ([HTTP://WWW.BAELDUNG.COM/PRIVACY-POLICY](http://www.baeldung.com/privacy-policy))

EDITORES ([HTTP://WWW.BAELDUNG.COM/EDITORS](http://www.baeldung.com/editors))

KIT DE MEDIOS (PDF) ([HTTPS://S3.AMAZONAWS.COM/BAELDUNG.COM/BAELDUNG+-+MEDIA+KIT.PDF](https://s3.amazonaws.com/baeldung.com/baeldung+-+media+kit.pdf))

