



Home » Java » Enterprise Java » How to use profiles in Spring Boot Application

ABOUT GAURAV BHARDWAJ



Gaurav has done Masters in Computer Applications(MCA) and is working in Software development field for more than 10 years in Java/J2EE technologies. He is currently working with one of top MNC. He has worked on various frameworks like Struts, Spring, Spring Boot, Angular JS, JSF, Velocity, iBatis, MyBatis, Hibernate, JUnit, Mockito, Dozer. He likes to explore new technologies and share his thoughts by writing a technical blog. He is the founder of JavaSolutionsGuide.blogspot.com.



How to use profiles in Spring Boot Application

Publicado por: Gaurav Bhardwaj en Enterprise Java 14 de julio de 2019 0 1756 vistas

Hola amigos,

En este tutorial, aprenderemos cómo usar los perfiles en una aplicación Spring Boot.



Vamos a discutir los siguientes puntos en este tutorial:

1. Qué es el Perfil de Spring Boot y por qué necesitamos perfiles
2. Cómo hacer un perfil en Spring Boot con el Ejemplo
3. Cómo configurar / cambiar el perfil predeterminado
1. Qué es el Perfil de Spring Boot y por qué necesitamos perfiles

HOJA INFORMATIVA

¡Los iniciados ya están disfrutando de actualizaciones semanales y librerías de código abierto! Únase a ellos ahora para obtener las últimas noticias en el mundo como a información sobre Android y otras tecnologías relacionadas.

☐ Estoy de acuerdo con los Términos y Condiciones y la Política de Privacidad

[Regístrate](#)

ÚNETE A NOSOTROS



Con **1,500** únicos miembros en **500** países ubicados en todos los sitios de Java. Con la ayuda de nuestra comunidad, busca de animar a nuestros miembros a escribir un blog con contenido único e interesante y consultar nuestro programa de escritura. También puedes ser un escritor en Java Code Geeks y perfeccionar tu escritura!

DEV, confirmar el código y desplegarlo en DEV, pero el problema ahora es que este código se conectará bien con la base de datos de DEV, pero cuando intente ejecutar este código de forma local, no funcionará, porque ha cambiado los detalles de la base de datos a la base de datos DEV.

Entonces, nuevamente, para que funcione en su local, tendrá que hacer los cambios en application.properties que son necesarios para local y ejecutar la aplicación.

Como puede ver, hay mucho ajeteo involucrado aquí en barajar entre el local y DEV.

Ahora imagine que tiene más entornos como ST, ET (QA), PROD y tiene que hacer cambios manualmente todas las veces. Será una verdadera pesadilla.

¿Entonces, cuál es la solución?

Perfiles de arranque de primavera en el rescate!

Spring Boot le permite externalizar la configuración de su aplicación para que pueda trabajar con el mismo código de aplicación en diferentes entornos sin que tenga que realizar cambios.

Spring Boot Profiles le permite configurar múltiples archivos application.properties, por entorno, de modo que cuando esté en local, utilizará el archivo de propiedades locales, cuando esté en DEV, utilizará el archivo de propiedades DEV y así sucesivamente, sin usted como un El programador necesita hacer cambios explícitos en el código.

Entonces, en general, si tiene algunas propiedades de aplicación que varían según el entorno, puede manejar eso con la ayuda de Spring Profiles.

Se ve bien. No es :)

2. ¿Cómo hacer perfiles en Spring Boot con ejemplos?

2.1 Siga mi publicación Cómo crear un proyecto Spring Boot con Spring Initializer y crear un proyecto Spring Boot con el nombre "Springbootprofiles". Agregue solo la dependencia web, ya que eso será suficiente para nuestras pruebas.

2.2 En el archivo .properties de la aplicación que ha creado automáticamente el inicializador de Spring, agregue la siguiente línea: application.environment = Este es un entorno local

2.3 Ejecute la aplicación haciendo clic en el proyecto y seleccionando Ejecutar como -> Ejecutar configuraciones -> Ejecutar

2.4 Verifique los registros de la consola generados por Spring Boot.

Verás la siguiente línea en los registros.

2019-07-07 20: 00: 52.147 INFO 15560 - [main] cbjsSpringbootprofilesApplication: No hay un conjunto de perfiles activo, volviendo a los perfiles predeterminados: predeterminado

Lo que básicamente indica que no hemos establecido ningún perfil explícitamente, por lo que Spring Boot usa un perfil predeterminado, lo que en otras palabras significa que Spring Boot usa configuraciones del archivo application.properties.

¿Cómo podemos comprobar eso?

Veamos en los próximos pasos.

2.5 Cree un controlador con el nombre ProfileController.java como seguimientos:

```

01 package com.blogspot.javasolutionsguide.springbootprofiles.controller;
02
03 import org.springframework.beans.factory.annotation.Value;
04 import org.springframework.web.bind.annotation.GetMapping;
05 import org.springframework.web.bind.annotation.RequestMapping;
06 import org.springframework.web.bind.annotation.RestController;
07
08 /**
09  * @author javaSolutionsGuide
10  */
11
12 @RequestMapping("/v1")
13 @RestController
14 public class ProfileController {
15
16     @Value("${application.environment}")
17     private String applicationEnv;
18
19     @GetMapping
20     public String getApplicationEnv() {
21         return applicationEnv;
22     }
23
24 }
```

Básicamente, lo que intentamos hacer es intentar acceder a la propiedad application.environment definida en el archivo application.properties en nuestro controlador mediante la anotación @Value, de modo que cuando lleguemos a este recurso uri desde el navegador, deberíamos obtener "Este es un ambiente local".

2.6 Comencemos de nuevo la aplicación haciendo clic en el proyecto y seleccionando Ejecutar como -> Ejecutar configuraciones -> Ejecutar y luego golpear el URI del recurso (<http://localhost:8080/v1>) y ver si está devolviendo el valor esperado del archivo application.properties.

Nota: Si lo desea, también puede cambiar el perfil predeterminado a algún otro archivo de propiedades. Lo veremos más adelante en este tutorial. 2.7 Ahora diga que desea implementar su código en el entorno DEV, de modo que desea que su aplicación elija propiedades específicas de DEV cuando la aplicación se ejecuta en DEV y las propiedades específicas del entorno LOCAL cuando la aplicación se ejecuta en local.

Para eso, lo que debemos hacer es crear otro archivo de propiedades con el nombre application-dev.properties. En la convención de nomenclatura general es application- {nombre de perfil} .properties '



Microservice Architecture - C started today.

Ad Build Microservices via our M service. Simple and efficient!

cloudamqp.com

Learn more

Donde el nombre del perfil es generalmente el nombre del entorno, pero puede ser cualquier texto.

2.8 Agreguemos la siguiente línea en el archivo application-dev.properties application.environment = Este es un entorno dev

2.9 Ahora, cómo decirle a la aplicación que use el perfil de desarrollo en lugar del perfil predeterminado.

Para eso, necesitamos establecer la variable de entorno "spring.profiles.active" como se muestra a continuación:

spring.profiles.active = dev

Para configurarlo, haga clic con el botón derecho en el proyecto, seleccione Ejecutar como -> Ejecutar configuraciones-> Entorno-> Nuevo -> Agregar nombre como spring.profiles.active y Valor como dev -> haga clic en ok -> Ejecutar

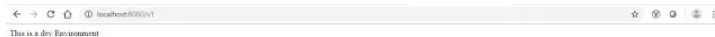
2.10 Si verificará los registros ahora, encontrará la siguiente línea en los registros:

2019-07-07 20: 22: 08.557 INFO 17172 - [main] cbjsSpringbootprofilesApplication: Los siguientes perfiles están activos: dev

Lo que muestra que el perfil de desarrollo está activo ahora.

2.11 Probemos realmente y veamos si nuestro controlador elige el valor de application-dev.properties

2.12 Haga clic en el recurso URI (http: // localhost: 8080 / v1) y vea el resultado en el navegador



Y está claro que, este valor de tiempo se ha seleccionado del archivo application-dev.properties.

Para llevar: Podemos tener n número de archivos de propiedades en nuestra aplicación Spring Boot para n número de entornos, que tendrán configuraciones específicas para ese entorno. Todo lo que tenemos que hacer para usar el archivo de propiedades en el entorno respectivo es establecer spring.profiles La propiedad .active para ese entorno y Spring Boot seleccionará el archivo de propiedades respectivo.

3. Cómo establecer / cambiar el perfil predeterminado

Como vimos anteriormente, de forma predeterminada, Spring Boot elige el perfil predeterminado, lo que significa que elige el archivo application.properties. Lo que sí, en cambio, queremos hacer de dev como nuestro perfil predeterminado.



En ese caso, todo lo que debe hacer es eliminar spring.profiles.active y establecer la propiedad spring.profiles.default en el perfil que queremos establecer como perfil predeterminado como variable de entorno en eclipse.

spring.profiles.default = dev

Ahora, si va a volver a ejecutar su aplicación, verá la siguiente línea en los registros de la consola:

2019-07-07 20: 25: 22.587 INFO 16922 - [main] cbjsSpringbootprofilesApplication: No hay un conjunto de perfiles activo, volviendo a los

del archivo `application.properties`. Si desea realizar cualquier otra configuración de entorno como predeterminada, puede establecer la propiedad `spring.profiles.default` en ese `environment` y

spring boot seleccionarán esa propiedad específica del entorno incluso cuando `spring.profiles.active` no esté configurado.

Gracias por leer. **Compártalo** con alguien que esto podría ser útil.

Publicado en Java Code Geeks
perfiles en la aplicación Spring

Las opiniones expresadas por

¡AL AGUA, GIGAS!

Etiquetado con: **SPRING BOOT**

(0 rating, 0 votes)

Debes ser un miembro registrado para calificar esto. [Iniciar la discusión](#) [1756 Vistas](#) [Tweet it!](#)

¿Quieres saber cómo desarrollar tu conjunto de habilidades para convertirte en un Java Rockstar?



Suscríbete a nuestro boletín para comenzar Rocking

Para comenzar, te regalamos nuestros libros electrónicos más vendidos **GRATIS!**

1. Mini libro de JPA
2. Guía de resolución de problemas de JVM
3. Tutorial JUnit para Pruebas Unitarias
4. Tutorial de anotaciones de Java
5. Preguntas de la entrevista de Java
6. Preguntas de la entrevista de primavera
7. Diseño de la interfaz de usuario de Android

y muchos más

☐ Estoy de acuerdo con los Términos y la Política de Privacidad

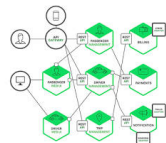
[Regístrate](#)

¿TE GUSTA ESTE ARTÍCULO? LEER MÁS DE JAVA CODE GEEKS



#1 VPN Service for Spain

Ad [expressvpn.com](#)



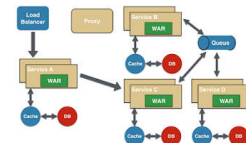
10 Tips to Become a Better Java Developer in 2018

[javacodegeeks.com](#)



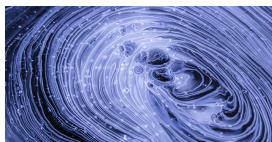
Industria 4.0

Ad [SoftExpert Software](#)



Microservice Design Patterns

[javacodegeeks.com](#)



Data Science Bootcamp IE

Ad [IE Business School](#)



Spring Boot Microservices , Docker and Kubernetes...

[javacodegeeks.com](#)



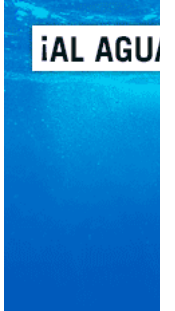
Microservices Interview Questions and Answers

[javacodegeeks.com](#)



Top 20 Libraries APIs Java Developers should know

[javacodegeeks.com](#)



Este sitio utiliza Akismet para reducir el spam. [Aprenda cómo se procesan los datos de sus comentarios](#) .

☒ Suscribirse ▼

BASE DE CONOCIMIENTOS

[Los cursos](#)

[Ejemplos](#)

[Minibooks](#)

[Recursos](#)

[Tutoriales](#)

FOGONADURA

[Mkyong](#)

LA RED DE CODE GEEKS

[Código .NET Geeks](#)

[Geeks de código de Java](#)

[Código de sistema Geeks](#)

[Código web Geeks](#)

SALÓN DE LA FAMA

[Serie "Tutorial completo de aplicaciones para Android"](#)

[11 sitios web de aprendizaje en línea que deberías visitar](#)

[Ventajas y desventajas de la computación en la nube - Pros y contras de la computación en la nube](#)

[Tutorial de Android Google Maps](#)

[Android JSON Parsing with Gson Tutorial](#)

[Android Location Based Services Application – GPS location](#)

[Android Quick Preferences Tutorial](#)

[Difference between Comparator and Comparable in Java](#)

[GWT 2 Spring 3 JPA 2 Hibernate 3.5 Tutorial](#)

[Java Best Practices – Vector vs ArrayList vs HashSet](#)

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on ultimate Java to Java developers resource center; targeted at the technical team lead (senior developer), project manager and junior developers. JCGs serve the Java, SOA, Agile and Telecom communities with daily news, domain experts, articles, tutorials, reviews, announcements, code snippets and source projects.

DISCLAIMER

Todas las marcas comerciales y marcas registradas que aparecen en Java Code Geeks son propiedad de sus respectivos dueños. Java es una marca comercial registrada de Oracle Corporation en los Estados Unidos y otros países. Ejemplos Java Code Geeks no está conectado a Oracle Corporation y no es propiedad de Oracle Corporation.