netmind
a bts company

Spring Boot

Spring Boot 3

Nuevas características y mejoras de Spring 6

Después de 5 años con Spring Framework 5, la nueva versión nos trae importantes cambios tales como:

- **Java 17**: la nueva versión de Spring Boot y del Framework obligaran a hacer uso de Java 17.
- **Sustituir Java EE por Jakarta EE:**
  - La versión mínima admitida es Jakarta EE9, lo que rompe compatibilidades con versiones anteriores.
  - El paquete javax.* migra a jakarta.*
  - Hará Spring Framework 6 compatible con las últimas versiones de servidores como Tomcat 10.1, Jetty 11 y Undertow 2.3.
- **Simplificación de Queries** con Spring Data.
- **Errores HTTP** más manejables y entendibles con la RFC 7807.
- **Compilación nativa**.
- **Observabilidad**: Ya que la tendencia es ir hacia un entorno más cloud native, se añade observabilidad basada en registro de métricas con Micrometer y proveedores como OpenZipkin y OpenTelemetr.y
- **JPMS**: Aunque no esta en la versión inicial la idea es tenerla para futuras versiones, un acceso más estricto al código de las aplicaciones y bibliotecas
- **Virtual Threads**: Spring 6 también incorporará características del proyecto Loom.
- Actualización a **Kotlin 1.7**, **Hibernate ORM 6.1**, **Jackson 2.14** y **R2DBC 1.0** para uso con Bases de Datos reactivas.
- Nueva versión de Flyway.

SpringBoot 3

- Spring Boot 3 es la nueva versión de Spring Boot que se integra con Spring 6.
- Esto implica una serie de cambios importantes a nivel de nuestros desarrollos.
- El cambio más importante viene dado por **Jakarka EE**.
- https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0-Release-Notes

Nuevas características Spring Boot 3

JDK 17

- Si bien antes ya existía soporte para Java 17, esta versión LTS ahora es la base.
- Al migrar desde la versión 11 de LTS, los desarrolladores de Java se beneficiarán de las nuevas funciones del lenguaje. Dado que Java en sí no es el tema de este artículo, solo nombraremos las características nuevas más importantes para los desarrolladores de Spring Boot.

Registros

- Los registros Java (JEP 395 , ver Java 14 Record Keyword) estaban pensados para usarse como una forma rápida de crear clases de soporte de datos, es decir, las clases cuyo objetivo es simplemente contener datos y transportarlos entre módulos, también conocidos como POJO (Plain Old Java Objets) y DTO (Data Transfer Objects).
- Podemos crear fácilmente DTO inmutables:

```
public record Person (String name, String address) {}
```

- Actualmente, debemos tener cuidado al combinarlos con Bean Validation porque las restricciones de validación no se admiten en los argumentos del constructor, como cuando la instancia se crea en la deserialización JSON (Jackson) y se coloca en el método de un controlador como parámetro.

Nuevas características Spring Boot 3

Bloques de texto

- Con JEP 378 , ahora es posible crear bloques de texto de varias líneas sin la necesidad de concatenar cadenas en saltos de línea:

```
String textBlock = """
Hello, this is a
multi-line
text block.
""";
```

Expresiones switch

- Java 12 introdujo expresiones de cambio (JEP 361), que (como todas las expresiones) evalúan un valor único y se pueden usar en declaraciones. En lugar de combinar operadores anidados *if – else (?:*), ahora podemos usar un *switch – case* -construct:

```
DayOfWeek day = DayOfWeek.FRIDAY;
int numOfLetters = switch (day) {
   case MONDAY, FRIDAY, SUNDAY -> 6;
   case TUESDAY -> 7;
   case THURSDAY, SATURDAY -> 8;
   case WEDNESDAY -> 9;
};
```

Pattern Matching

- Las coincidencias de patrones se elaboraron en el Proyecto Amber y llegaron al lenguaje Java. En el lenguaje Java, pueden ayudar a simplificar el código, por *ejemplo,* de evaluaciones.

```
if (obj instanceof String s) {
    System.out.println(s.toLowerCase());
}
```

Nuevas características Spring Boot 3

Clases e interfaces selladas

- Las clases selladas pueden limitar la herencia especificando subclases permitidas:
- Más detalles en [Clases e interfaces selladas en Java](#) .

```
public abstract sealed class Pet permits Dog, Cat {}
```

Yakarta EE 9

- El cambio más importante podría ser el salto de Java EE a Jakarta EE9, donde el espacio de nombres del paquete cambió de *javax.** a *jakarta.** . Como resultado, necesitamos ajustar todas las importaciones en nuestro código siempre que usemos clases de Java EE directamente.
  - Por ejemplo, cuando accedemos al objeto *HttpServletRequest* dentro de nuestro controlador Spring MVC, debemos reemplazar:
- También debemos ser conscientes de esto cuando usamos bibliotecas externas que dependen de Java/Jakarta EE (por ejemplo, tenemos que usar Hibernate Validator 7+, Tomcat 10+ y Jetty 11+).

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletRequest;
```

Nuevas características Spring Boot 3

Otras dependencias

- Spring Framework 6 y Spring Boot 3 necesitan las siguientes versiones mínimas:
- Kotlin 1.7+
- Lombok 1.18.22+ (**soporte JDK17**)
- Gradle 7.3+

Cambios pequeños en Spring Web MVC

- Una de las nuevas características más importantes es la compatibilidad con RFC7807 (Estándar de detalles de problemas). Ahora no necesitaremos incluir bibliotecas separadas, como Zalando Problem .
- Otro cambio menor es que HttpMethod ya no es una enumeración, sino una clase que nos permite crear instancias para métodos HTTP extendidos, por ejemplo, aquellos definidos por WebDAV:

```
HttpMethod lock = HttpMethod.valueOf("LOCK");
```

- Al menos algunas integraciones obsoletas basadas en servlets se eliminan, como Commons FileUpload (deberíamos usar *StandardServletMultipartResolver* para cargas de archivos de varias partes), Tiles y compatibilidad con FreeMarker JSP (deberíamos usar vistas de plantilla de FreeMarker en su lugar).

Nuevas características Spring Boot 3

Puntos importantes

Se ha prestado especial atención a dos temas generales: los ejecutables nativos y la observabilidad . General significa que:

- Spring Framework introduce **abstracciones core**.
- Los **proyectos de portfolio** se integran consistentemente con ellos.
- Spring Boot proporciona **configuración automática**.

1. Ejecutables nativos

- Crear ejecutables nativos e implementarlos en GraalVM tiene una mayor prioridad. Por tanto, la iniciativa Spring Native se está trasladando a Spring propiamente dicha .
- Para la generación de AOT, no es necesario incluir complementos separados, simplemente podemos usar un nuevo goal de *spring-boot-maven-plugin:*:

```
mvn spring-boot:aot-generate
```

- Native Hints también será parte del núcleo de Spring. La infraestructura de prueba para esto estará disponible con Milestone 5 (v6.0.0-M5) .

Nuevas características Spring Boot 3

Puntos importantes

2. Observabilidad

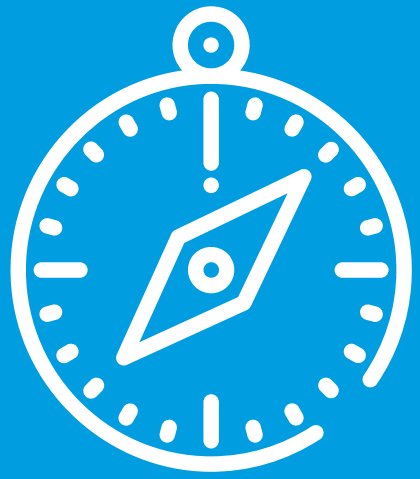
- Spring 6 presenta **Spring Observability**, una nueva iniciativa que se basa en Micrometer y Micrometer Tracing (anteriormente Spring Cloud Sleuth). El objetivo es registrar de manera eficiente las métricas de las aplicaciones con Micrometer e implementar el seguimiento a través de proveedores, como OpenZipkin u OpenTelemetry .
  - https://spring.io/blog/2022/10/12/observability-with-spring-boot-3?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=en&_x_tr_pto=wapp

- Hay configuración automática para todos estos en Spring Boot 3, y los proyectos de Spring están trabajando para instrumentarse utilizando la nueva API de Observación.
- Más detalles en el artículo: https://www.baeldung.com/spring-boot-3-observability

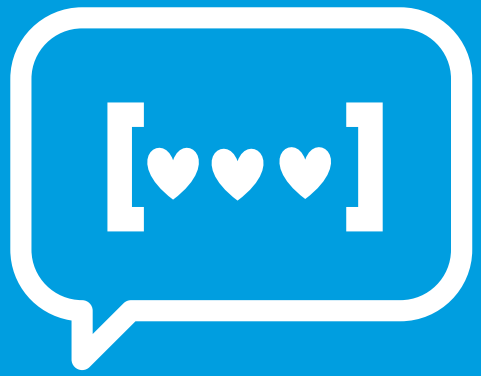
Proyectos de migración

Hay algunos consejos para la migración de proyectos que debemos conocer. Los pasos recomendados son:

1. Migrar a **Spring Boot 2.7** (si tenemos una versión anterior)
   - Guía de migración: https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-3.0-Migration-Guide
2. Verificar el uso de código obsoleto y el procesamiento de archivos de configuración heredados ; Se eliminará con la nueva versión principal.
3. Migrar a Java 17.
4. Verificar que los proyectos de terceros tengan versiones compatibles con Jakarta EE 9.



Next steps



We would like to know your opinion!

Please, let us know what you think about the content.
From Netmind we want to say thank you, we appreciate time and effort you have taking in answering all of that is important in order to improve our training plans so that you will always be satisfied with having chosen us
quality@netmind.es

netmind
a bts company

Thanks!

Follow us: