Vorbereitung auf Spring Boot 3

Spring Boot 2.0 war die erste Version in der 2.x-Reihe und wurde am 28. Februar 2018 veröffentlicht. Wir haben gerade Spring Boot 2.7 veröffentlicht, was bedeutet, dass wir die 2.x-Reihe bisher etwas mehr als 4 Jahre lang beibehalten haben Jahre. Insgesamt haben wir in diesem Zeitraum 95 verschiedene Veröffentlichungen veröffentlicht! Das gesamte Spring-Team und viele in unserer Community von Mitwirkenden bereiten sich jetzt auf die nächste Spring-Generation vor. Wir planen die Veröffentlichung von Spring Boot 3.0 im November 2022. Diese nächste große Überarbeitung basiert auf Spring Framework 6.0 und erfordert Java 17 oder höher. Es wird auch die erste Version von Spring Boot sein, die Jakarta EE 9-APIs (jakarta.*) anstelle von EE 8 (javax.*) verwendet. Die nächsten sechs Monate bieten eine ideale Gelegenheit, Ihre eigenen Projekte auf dieses Major Release vorzubereiten. In diesem Blogbeitrag behandeln wir einige der Dinge, die Sie heute tun können , um eine zukünftige Migration so einfach wie möglich zu gestalten.

Aktualisieren Sie auf Java 17

Spring Boot 3.0 erfordert Java 17, aber Sie müssen nicht bis zu dieser Version warten, um auf die neueste LTS-Java-Version zu aktualisieren. Alle neueren Versionen von Spring Boot 2.x funktionieren sehr gut mit Java 17. Sie können auch Java 17-Funktionen (z. B. Datensätze) in Ihren eigenen Codebasen verwenden. Wir empfehlen Ihnen dringend, Ihr JDK nach Möglichkeit noch heute zu aktualisieren.

Führen Sie ein Upgrade auf den neuesten Spring Boot 2.7.x durch

Wenn Sie derzeit eine ältere Version von Spring Boot 2.x verwenden, empfehlen wir Ihnen dringend, auf Spring Boot 2.7 zu aktualisieren. Wenn Spring Boot 3.0 veröffentlicht wird, werden wir einen Migrationsleitfaden bereitstellen, der jedoch davon ausgeht, dass Sie von Spring Boot 2.7 und nicht von einer früheren Version migrieren.Upgrade-Anweisungen finden Sie immer in unseren Versionshinweisen. Wenn Sie beispielsweise von Spring Boot 2.6 auf Spring Boot 2.7 aktualisieren, können Sie diesem Abschnitt folgen .Wenn Sie von Spring Boot 2.5 oder früher aktualisieren, empfehlen wir nicht, Releases zu überspringen. Es ist oft einfacher, schrittweise zu aktualisieren (z. B. 2.5 . 2.6 . 2.7), als zu versuchen, direkt von 2.5 . 2.7 zu aktualisieren.

Suchen Sie nach Aufrufen von veraltetem Code

Im Zuge der Weiterentwicklung von Spring Boot werden wir häufig Methoden oder Klassen verwerfen und Ersatz bereitstellen. Wir bieten in der Regel eine 12-monatige Überlappung, nach der veralteter Code entfernt wird.Spring Boot 3.0 entfernt den gesamten veralteten Code, daher empfehlen wir Ihnen, zu überprüfen, ob Ihr vorhandener Code auf veralteten Methoden beruht. Es ist erwägenswert, die -WerrorJava-Compiler-Option zu verwenden, um Ihren Build fehlschlagen zu lassen, wenn Verfallswarnungen gemeldet werden.

Migrieren Sie von Legacy application.properties und application.yaml Processing

Spring Boot 2.4 hat die Art und Weise geändert, application.propertieswie application.yamlDateien geladen wurden. Die meisten Benutzer haben die Änderung nicht bemerkt, aber einige Projekte haben die spring.config.use-legacy-processingEigenschaft möglicherweise auf gesetzt true, um das alte Verhalten wiederherzustellen.Die Legacy-Verarbeitungsunterstützung wird in Spring Boot 3.0 nicht verfügbar sein, daher sollten Sie überprüfen, ob Ihr Projekt nicht festgelegt ist spring.config.use-legacy-processing.

Verwenden Sie den PathPatternParser von Spring MVC

Spring MVC bietet zwei Möglichkeiten, Muster zu analysieren. Ab Spring Boot 2.6 PathPatternParserwird standardmäßig der verwendet.Einige Anwendungen haben möglicherweise manuell zur AntPathMatcherImplementierung zurückgeschaltet, indem Sie die spring.mvc.pathmatch.matching-strategyEigenschaft festgelegt haben. Obwohl dies in Spring Boot 3.0 weiterhin funktioniert, empfehlen wir, PathPatternParserwenn möglich, die zu verwenden, da sie eine bessere Leistung bietet.

Überprüfen Sie, ob Projekte von Drittanbietern Jakarta EE 9-kompatible Versionen haben

Jakarta EE 9 ist ein neues Top-Level- jakartaPaket, das das Top-Level-Paket von EE 8 ersetzt javax. Beispielsweise verwendet die Servlet-Spezifikation in Jakarta EE 8 ein javax.servletPaket, aber dies wurde jakarta.servletin EE 9 geändert.Im Allgemeinen ist es nicht möglich, Java EE- und Jakarta EE-APIs im selben Projekt zu mischen. Sie müssen sicherstellen, dass Ihr eigener Code sowie alle Bibliotheken von Drittanbietern jakarta.*Paketimporte verwenden.Die gute Nachricht ist, dass die meisten gut gewarteten Bibliotheken Jakarta EE 9-kompatible Varianten produzieren. Hibernate, Thymeleaf, Tomcat, Jetty und Undertow haben dies beispielsweise bereits getan.Wir empfehlen Ihnen, sich etwas Zeit zu nehmen, um zu überprüfen, ob alle von Ihnen verwendeten Bibliotheken von Drittanbietern in Jakarta EE integriert sind, und überprüfen Sie, ob sie EE 9-kompatible Varianten haben. Die häufigsten Probleme, die wir gefunden haben, treten bei Projekten auf, die Servlet-APIs importieren.

Überprüfen Sie, ob Projekte von Drittanbietern aktualisierte frühlingskompatible Versionen haben

Spring Framework 6.0 ist nicht binärkompatibel mit der vorherigen Generation. Wenn Sie Jars von Drittanbietern verwenden, die eine Spring-Integration anbieten, sollten Sie überprüfen, ob sie mit Spring Framework 6 kompatible Versionen planen.

Probieren Sie die Spring Boot-Meilensteine ..aus

Obwohl wir es nicht für die Produktion empfehlen, können Sie die Meilensteine ..von Spring Boot 3.0 noch heute ausprobieren, um zu sehen, wie schwierig es sein wird, Ihr Projekt zu migrieren. Das Ausprobieren der Meilensteine ..in einem Zweig Ihres Codes ist eine großartige Möglichkeit, potenziellen Problemen vorzubeugen. Sie sind unter https://repo.spring.io/milestone verfügbar . Durch das Erstellen eines Projekts auf https://start.spring.io und die Auswahl des neuesten 3.0-Meilensteins werden die erforderlichen Repositories in die Build-Konfiguration aufgenommen.Wir sind immer an Feedback interessiert und freuen uns sehr, wenn Fehler identifiziert werden, bevor wir die GA-Version veröffentlichen.Sie können Probleme unter github.com/spring-projects/spring-boot/issues melden (stellen Sie sicher, dass Sie uns die Version von Spring Boot mitteilen, die Sie verwenden).

Feedback

War dieser Blog für Sie wertvoll. Wir danken für jede Anregung und Feedback

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/servletin