

Was ist Java Spring Boot?

Eine Einführung in Spring Boot – das beliebte Java-basierte Tool zum Entwickeln von Web-Apps und Microservices.

Ressourcen▼



Was ist Spring Boot?

Java Spring Boot ist ein Open-Source-Tool, das die Verwendung von Java-basierten Frameworks zum Erstellen von Microservices und Web-Apps vereinfacht. Bei der Definition von Spring Boot muss zunächst Java erwähnt werden – eine der beliebtesten und meistverbreiteten Entwicklungssprachen und Computingplattformen für die App-Entwicklung. Entwickler auf der ganzen Welt beginnen ihre Journey in die Programmierung, indem Sie Java erlernen. Java ist flexibel und benutzerfreundlich und wird von Entwicklern gerne für eine Vielzahl von Apps eingesetzt – von Social Media-, Web- und Gaming-Apps bis hin zu Netzwerk- und Unternehmensanwendungen.

Java und das Spring-Framework

Java ist vielleicht einfach in der Verwendung und leichter zu erlernen als andere Programmiersprachen, allerdings ist das Erstellen, Debuggen und Bereitstellen von Java-Apps extrem komplex geworden. Dies liegt an der exponentiellen Zunahme von Variablen, mit denen moderne Fachkräfte in der Entwicklung konfrontiert sind, wenn sie Web-Apps oder mobile Apps für gängige moderne Technologien wie Musikstreaming oder mobile Barzahlungs-Apps entwickeln. Ein Entwickler, der eine einfache branchenspezifische App programmiert, muss sich jetzt mit mehreren Bibliotheken, Plug-Ins, Bibliotheken für die Fehlerprotokollierung- und -behandlung, Webdienstintegrationen und Programmiersprachen wie C#, Java, HTML usw. befassen. Es ist verständlich, dass es eine immense Nachfrage nach Tools gibt, die die Entwicklung von Java-Apps optimieren und Entwicklern Zeit und Geld sparen.

Nutzen Sie Anwendungsframeworks, die große Mengen an vorprogrammiertem Code bieten, den Entwickler je nach Bedarf verwenden und ihrem eigenen Code hinzufügen können. Diese Frameworks entlasten Entwickler bei fast allen Anforderungen – egal, ob sie mobile und Web-Apps entwickeln oder mit Desktops und APIs arbeiten. Frameworks sorgen dafür, dass Apps schneller, einfacher und sicherer erstellt werden können, indem wiederverwendbarer Code und Tools bereitgestellt werden, die helfen, die verschiedenen Elemente eines Softwareentwicklungsprojekts miteinander zu verknüpfen.

Hier kommt Spring ins Spiel: Spring ist ein Open-Source-Projekt, das einen optimierten, modularen Ansatz zum Erstellen von Apps mit Java bietet. Die Familie der [Spring-Projekte](#) entstand im Jahr 2003 als Reaktion auf die Komplexität der frühen Java-Entwicklung und bietet Unterstützung bei der Entwicklung von Java-Apps. Der Name Spring bezieht sich in der Regel auf das Anwendungsframework selbst oder aber auf die gesamte Gruppe von Projekten oder Modulen. Java Spring Boot ist ein spezifisches Modul, das als Erweiterung des Spring-Frameworks erstellt wurde.

Mit diesem Hintergrundwissen über das Zusammenspiel von Spring-Framework, Spring Boot und Java folgt nun die Definition von Spring Boot: Ein Tool, das die Entwicklung von Web-Apps und Microservices innerhalb des Java-Frameworks mit dem Namen Spring optimiert und beschleunigt.

Da die Programmiersprache und/oder Plattform „Java“ lautet, wird die Lösung manchmal auch "Java Spring Boot", "Java Spring-Framework" oder "Spring Boot-Framework" genannt. Da es jedoch andere Java-Frameworks von Drittanbietern gibt, z. B. Play und Hibernate, ist es genauer, von "Spring" und "Spring Boot" zu reden.

Eine allgemeine Erläuterung der Java-Grundlagen finden Sie im Artikel [Was ist Java?](#)

Worin liegt der Unterschied zwischen Spring und Spring Boot?

Wie bereits erwähnt, ist Spring ein Java-basiertes Open-Source-App-Framework, das viele kleinere Projekte vereint. Andere beliebte [Spring-Projekte](#) sind Spring Data, Spring Cloud und Spring Security, um nur einige zu nennen. Damit der Unterschied zwischen Spring Boot und Spring klar wird, ist es wichtig zu wissen, dass es zwar Unterschiede in den Kernfunktionen gibt, aber alle zur Spring-Familie gehören.

Um die Unterschiede zwischen Spring und Spring Boot vollständig zu verstehen, müssen wir zuerst einige wichtige Begriffe definieren.



Was sind Microservices?

[Microservices](#) sind ein Ansatz in der Softwareentwicklungsarchitektur. Das "Micro" in Microservices bezieht sich darauf, dass der Code in kleinen, handlichen Teilen oder Komponenten bereitgestellt wird und dass jeder "Dienst" (Service) oder jede Kernfunktion unabhängig von den anderen Diensten erstellt und bereitgestellt wird. Die Zusammenarbeit und Kommunikation der unabhängigen Komponenten erfolgt über vorgeschriebene API-Dokumente, die als Verträge bezeichnet werden. Der geringe Umfang und die relative Isolation dieser Microservices bieten viele Vorteile. Da diese Art von Architektur verteilt und lose gekoppelt ist, stürzt beispielsweise nicht die gesamte App ab, wenn eine Komponente ausfällt. Weitere Vorteile sind eine höhere Produktivität, einfachere Wartung, bessere Geschäftsausrichtung und höhere Fehlertoleranz.

Was ist Abhängigkeitsinjektion?

Die Abhängigkeitsinjektion (Dependency Injection, DI) ist eine Entwurfstechnik, mit der IoC (Inversion of Control, Steuerungsumkehr) erreicht werden soll. Bei der objektorientierten Programmierung beispielsweise mit Java werden Objekte, die von anderen Objekten abhängig sind, als Abhängigkeiten bezeichnet. In der Regel wird das empfangende oder abhängige Objekt als Client bezeichnet, und das Objekt, von dem der Client abhängig ist, wird als Dienst bezeichnet. Daher wird bei der Abhängigkeitsinjektion der Dienst an den Client übergeben oder die Abhängigkeit mithilfe von Code, der als Injektor bezeichnet wird, "injiziert". Dank DI muss der Client nicht angeben, welcher Dienst verwendet werden soll. Diese Aufgabe übernimmt stattdessen der Injektor für den Client.



Was ist Konvention vor Konfiguration?

„Konvention vor Konfiguration“, manchmal auch Coding by Convention genannt, ist ein Konzept, das in Anwendungsframeworks verwendet wird, um die Anzahl der Entscheidungen zu reduzieren, die ein Entwickler treffen muss. Es entspricht dem "DRY"-Prinzip (Don't Repeat Yourself) und soll helfen, redundanten Code zu vermeiden. Konvention vor Konfiguration sorgt dafür, dass die Flexibilität erhalten bleibt, und ermöglicht es dem Entwickler, nur Code für die Aspekte der zu erstellenden App zu schreiben, die nicht der Konvention entsprechen. Wenn das gewünschte Verhalten der App den festgelegten Konventionen entspricht, wird die App standardmäßig ausgeführt, ohne dass Konfigurationsdateien geschrieben werden müssen. Der Entwickler muss Konfigurationsdateien nur explizit schreiben, wenn das gewünschte Verhalten von der "Konvention" abweicht.

Spring-Framework im Vergleich zu Spring Boot

Da Spring Boot auf den Stärken des Spring-Frameworks aufbaut, stellt sich nicht wirklich eine "entweder/oder"-Frage von Spring oder Spring Boot. Vielmehr geht es um einen "sowohl/als auch"-Nutzen. Spring und Spring Boot konkurrieren nicht miteinander, sondern arbeiten Hand in Hand:

	Spring	Spring Boot
Was ist es?	Ein Open-Source-Webanwendungsframework, das auf Java basiert	Eine Erweiterung oder ein Modul, die bzw. das auf dem Spring-Framework basiert
Was bietet es?	Bietet eine flexible, vollständig konfigurierbare Umgebung mit Tools und Bibliotheken von vordefiniertem Code, um angepasste, lose gekoppelte Web-Apps zu erstellen	Bietet die Möglichkeit, eigenständige Spring-Anwendungen zu erstellen, die sofort ausgeführt werden können, ohne dass Anmerkungen oder XML-Konfigurationen erforderlich sind oder dass große Mengen an zusätzlichem Code geschrieben werden müssen
Wann sollte ich es verwenden?	Verwenden Sie Spring, wenn Sie Folgendes benötigen: <ul style="list-style-type: none">• Flexibilität• Einen flexiblen Ansatz*• Benutzerdefinierten Code ohne Abhängigkeiten• Eine sehr individuelle Konfiguration• Entwicklung von Unternehmensanwendungen	Verwenden Sie Spring Boot, wenn Sie Folgendes benötigen: <ul style="list-style-type: none">• Einfache Nutzung• Einen statischen Ansatz*• Qualitativ hochwertige Apps, die direkt einsatzfähig sind, und kürzere Entwicklungszeiten• Keine Programmierung von Codebausteinen oder Konfiguration von XML• Entwicklung von REST-APIs
Was ist die Hauptfunktion?	Abhängigkeitsinjektion	Autokonfiguration
Verfügt es über eingebettete Server?	Nein. In Spring müssen Sie die Server explizit einrichten.	Ja, Spring Boot verfügt über integrierte HTTP-Server wie Tomcat und Jetty.
Wie wird es konfiguriert?	Das Spring-Framework bietet Flexibilität, muss allerdings manuell konfiguriert werden.	Spring und andere Frameworks von Drittanbietern werden von Spring Boot automatisch gemäß dem Standardprinzip "Konvention vor Konfiguration" konfiguriert.
Muss ich dazu Kenntnisse in XML aufweisen?	In Spring sind Kenntnisse der XML-Konfiguration erforderlich.	Spring Boot erfordert keine XML-Konfiguration.
Gibt es CLI-Tools zum Entwickeln/Testen von Apps?	Das Spring-Framework selbst stellt keine CLI-Tools zum Entwickeln oder Testen von Apps bereit.	Spring Boot verfügt als Spring-Modul über ein CLI-Tool zum Entwickeln und Testen Spring-basierter Apps.
Funktioniert es nach einem statischen oder flexiblen Ansatz?	Flexibel*	Statisch*

*Statischer oder flexibler Framework-Ansatz

Auch wenn der Begriff "statisch" etwas negativ klingen mag, haben sowohl der statische Ansatz von Spring Boot als auch das flexible Spring-Framework ihre Vorteile.

Statischer Ansatz

Ein statischer Ansatz vertritt den Standpunkt, dass es **einen Weg** gibt, der wesentlich einfacher ist als alle anderen. Die Software schränkt Entwickler von vornherein ein und ermutigt sie, die Dinge auf die vorgeschriebene Weise zu tun.

Sie bietet einen gut geebneten Weg, also eine Best Practice, die für die meisten Menschen in den meisten Situationen funktioniert. Die App wird in enger Anlehnung an diese Best Practices und weitverbreiteten Konventionen geschrieben. Ein statischer Ansatz erleichtert nicht nur die Zusammenarbeit, sondern hilft auch, sich bei einem Programmierprojekt Unterstützung zu holen. Andere Entwickler, die Erfahrung mit diesem Framework haben, sind sofort mit der neuen App vertraut und können direkt mitarbeiten.

Flexibler Ansatz

Ein flexibler Ansatz wird verwendet, wenn alle Lösungen ungefähr denselben Aufwand erfordern oder dieselbe Komplexität aufweisen. Hier wird davon ausgegangen, dass es nicht **nur einen richtigen Weg** gibt, um eine Lösung für ein Problem zu finden. Stattdessen werden flexible Tools bereitgestellt, mit denen das Problem auf viele verschiedene Weisen gelöst werden kann. Flexible Frameworks haben den Vorteil, dass sie viel Flexibilität bei der Entwicklung bieten und Entwicklern mehr Kontrolle geben. Der größte Nachteil bei so viel Flexibilität besteht darin, dass der Entwickler mehr Entscheidungen treffen und am Ende vielleicht mehr Code schreiben muss, da das Framework so offen und flexibel ist.

Funktionen und Vorteile von Spring Boot

Das Spannende sind letztendlich die Kernfunktionen, die Java Spring Boot über das Spring-Framework hinaus bietet. Mit Spring Boot lassen sich Apps einfacher und schneller einrichten, konfigurieren und ausführen. So entfällt die aufwändige Konfiguration, die für die Einrichtung der meisten Spring-basierten Apps erforderlich ist. Entwickler können direkt loslegen und Spring Boot nutzen, ohne sich mit dem zugrunde liegende Spring-Framework auseinandersetzen zu müssen.



Hier erfahren Sie, wie Spring Boot-Funktionen helfen, Java-Programmieraufgaben zu optimieren. Einige wichtige Funktionen von Spring Boot:

- ✓ **Eigenständige Anwendungen:** Spring Boot hilft beim Erstellen von Apps, die nicht an eine bestimmte Plattform gebunden sind und lokal auf einem Gerät ausgeführt werden können, ohne dass eine Internetverbindung besteht oder andere installierte Dienste funktionsfähig sein müssen.
- ✓ **Eingebettete Server:** Mit Spring Boot können Sie Server wie Tomcat, Jetty oder Undertow direkt einbetten.
- ✓ **Statischer Ansatz:** Spring Boot vereinfacht Buildkonfigurationen, indem statische Starter-Abhängigkeiten bereitgestellt werden.
- ✓ **Automatische Konfiguration:** Spring und andere Drittanbieterbibliotheken werden von Spring Boot nach Möglichkeit automatisch konfiguriert.
- ✓ **Produktionsbereite Funktionen:** Spring Boot bietet produktionsbereite Funktionen wie Metriken, Integritätsprüfungen und externalisierte Konfigurationen.

Vorteile von Spring Boot

Spring Boot:

- ✓ **Verkürzt die Entwicklungszeit und erhöht die Produktivität:** Mit Spring Boot wird die Entwicklung Spring-basierter Apps mit Java erheblich vereinfacht. Der statische Ansatz des Spring-Frameworks verkürzt die Zeit für die Entscheidungsfindung und wiederholte Aufgaben, sodass mehr Zeit für das Erstellen und Testen der Apps verfügbar ist.
- ✓ **Verringert die Notwendigkeit, Codebausteine, Anmerkungen und XML-Konfigurationen zu schreiben:** Entwickler müssen weder Code generieren noch XML-Code konfigurieren. Sie müssen sich auch nicht mit dem Spring-Framework vertraut machen, wenn sie dies nicht möchten.
- ✓ **Integriert Apps in die Spring-Projektfamilie:** Spring Boot-Apps werden nahtlos in andere Projekte im Ökosystem des Spring-Frameworks – wie Spring Data, Spring Cloud, Spring Security – sowie in andere vertrauenswürdige Clouddienste wie Microsoft Azure Spring Cloud integriert.
- ✓ **Stellt Dev/Test-Tools bereit:** Das Spring Boot-CLI-Tool (Command-Line Interface) und eingebettete HTTP-Server machen es sehr einfach, Umgebungen zum Entwickeln/Testen Spring-basierter Apps zu erstellen.
- ✓ **Bietet Plug-Ins und Tools, um die Entwicklung zu vereinfachen:** Spring Boot bietet Plug-Ins für die Arbeit mit In-Memory-Datenbanken sowie andere beliebte Tools für die Buildautomatisierung wie Apache Maven.

Java Spring Boot-Tutorials

Wenn Sie bereits mit dem Erstellen von Apps mithilfe des Spring-Frameworks vertraut sind, wird es Ihnen wahrscheinlich leicht fallen, Spring Boot unter Verwendung der Tutorials, Projekt-Starter und Dokumentation in diesem Abschnitt zu erlernen. Aber es gibt gute Nachrichten, wenn Sie es eilig haben, Spring Boot zu verwenden, das Framework aber noch nicht kennen. Es ist möglich, Spring Boot zu erlernen – und direkt damit zu arbeiten – bevor Sie sich mit dem Spring-Framework auskennen, da Sie viele Grundlagen des Frameworks im Laufe der Zeit kennenlernen. Dabei erfahren Sie zwar nicht alles, was es über das Framework zu wissen gibt, allerdings erhalten Sie einen guten Einstieg, wenn Sie Spring später kennenlernen möchten.

Ein weiterer Vorteil, Spring Boot kennenzulernen, ist neben der schnelleren und effizienteren App-Entwicklung die Möglichkeit, andere Spring-Funktionen zu nutzen, nämlich Spring Data, Project Reactor und Spring Security.



Abhängigkeitsinjektion

- XML-basiert, d. h. <beans/> und <namespace:*/>
- Anmerkungsbasiert, d. h. @Component, @Autowired, @Inject
- Java-basiert, d. h. @Configuration-Klassen, @Bean-Methoden

Inversion of Control (IoC)

- ApplicationContext: die Stammschnittstelle, die als Spring-IoC-Container dient
- BeanFactory: die Schnittstelle für den Zugriff auf den Spring-Container
- Beans: die Objekte, die durch ApplicationContext instanziiert, assembliert und verwaltet werden

Java Spring Boot-Tutorials für Einsteiger

Sobald Sie die Grundlagen beherrschen und Spring Boot-Apps im großen Stil entwickeln und bereitstellen möchten, sollten Sie den vollständig verwalteten [Azure Spring Cloud](#) -Dienst von Microsoft und VMware in Betracht ziehen. Alternativ können Sie direkt mit dem Tutorial beginnen, in dem Sie erfahren, wie Sie [Spring Boot-Apps mit TLS-/SSL-Zertifikaten schützen](#).

Java-App erstellen und schützen

Erstellen Sie eine Java-App mit Spring Initializr unter Verwendung von Spring Boot Starter für [Microsoft Entra ID \(ehemals Azure AD\)](#).

[Jetzt starten](#) >

Erste App in Azure Spring Cloud bereitstellen

Erfahren Sie, wie Sie eine einfache Spring Boot-App entwickeln, bereitstellen und überwachen, die mit Spring Initializr erstellt wurde.

[Jetzt starten](#) >

Einfachen Spring Boot-Microservice erstellen

Erstellen Sie einen einfachen Microservice, und stellen Sie ihn bereit – und erfahren Sie, wie Sie ihm Spring Cloud-Technologien hinzufügen.

[Jetzt starten](#) >

Weitere Spring Boot Starter-Optionen

Erweitern Sie Ihre Spring Boot-Apps durch Funktionen für Speicher, Sicherheit, Authentifizierung und Schlüsseltresore.

[Jetzt starten >](#)

Häufig gestellte Fragen

[Alle erweitern](#) | [Alle reduzieren](#)

➤ Was ist Java Spring Boot?

> Was ist Spring Boot in Java?

- › Worin liegt der Unterschied zwischen Spring und Spring Boot?

- › Ist Spring Boot ein Back-End-System?

➤ Welche Vorteile bietet Spring Boot?

- › Wo finde ich Spring Boot-Tutorials für Einsteiger?

Sind Sie bereit für ein kostenloses Azure-Konto? Los geht's!

Azure kostenlos testen

Azure erkunden	Produkte und Preise	Lösungen und Support	Partner	Ressourcen	Cloud Computing
Was ist Azure?	Produkte	Lösungen	Azure Marketplace	Dokumentation	Was ist Cloud Computing?
Erste Schritte mit Azure	Preise von Azure	Ressourcen für schnelleres Wachstum	Einen Partner finden	Blog	Was ist eine Multicloud?
Globale Infrastruktur	Kostenlose Azure-Dienste	Lösungsarchitektur	ISV Success beitreten	Entwicklerressourcen	Was ist maschinelles Lernen?
Rechenzentren nach Regionen	Flexible Kaufoptionen	Support		Studierende	Was ist Deep Learning?
Vertrauen in die Cloud	FinOps in Azure	Demonstration von Azure mit Diskussionsgelegenheit		Veranstaltungen und Webinare	Was ist AlaaS?
Azure Essentials	Kosten optimieren			Analystenberichte, Whitepapers und E-Books	Was sind große Sprachmodelle (Large Language Models, LLMs)?
Kundenreferenzen				Videos	Was sind kleine Sprachmodelle (Small Language Models, SLMs)?
					Was ist Retrieval-Augmented Generation

(RAG)?



Deutsch (Deutschland)



Ihre Datenschutzoptionen

Verbraucherdatenschutz für Gesundheitsdaten

[An Microsoft wenden](#)

[Abo kündigen](#)

[Impressum](#)

[Datenschutz](#)

[Cookies verwalten](#)

[Nutzungsbedingungen](#)

[Markenzeichen](#)

[Informationen zu unserer Werbung](#)

[EU Compliance DoCs](#)

[© Microsoft 2025](#)