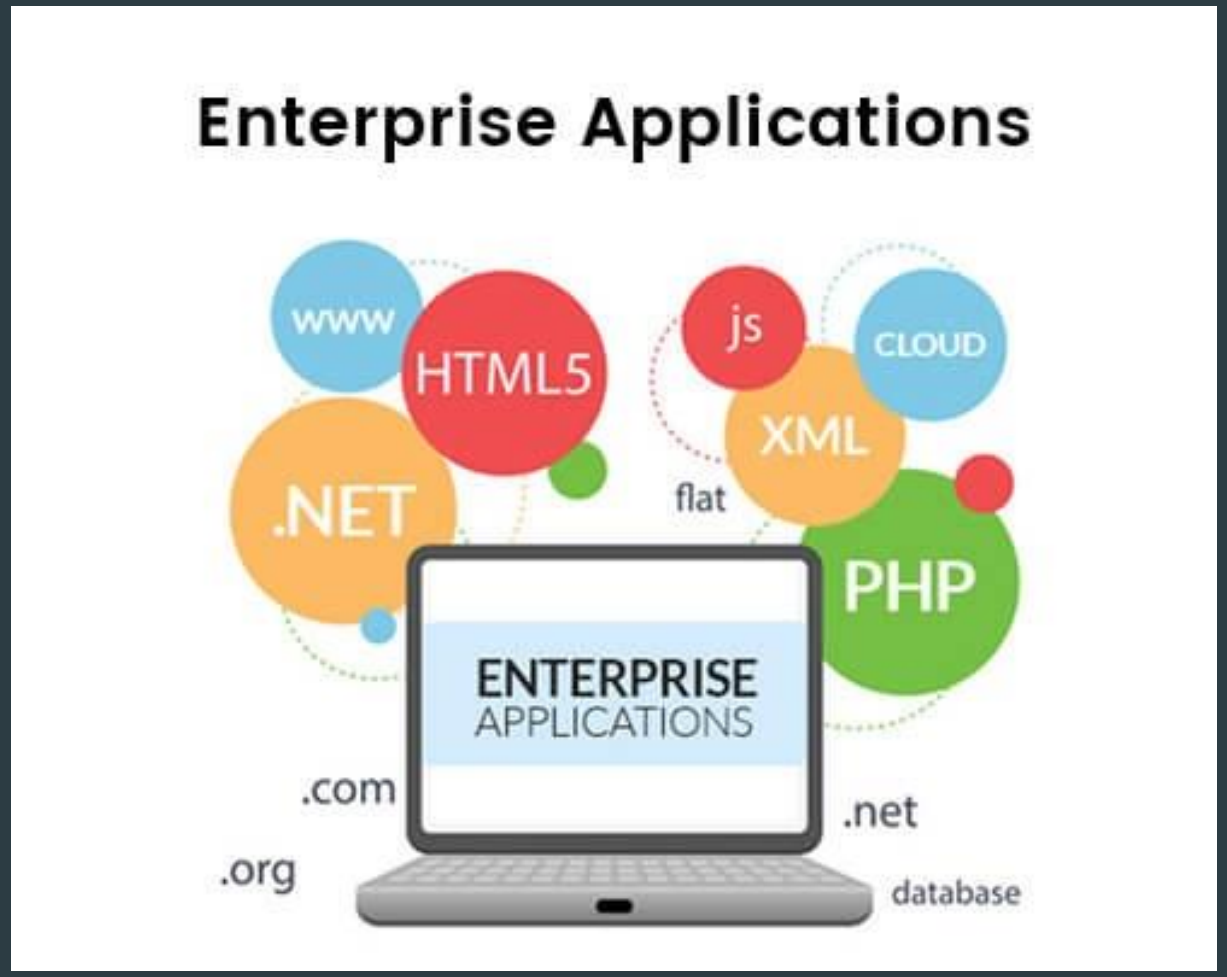
Java Enterprise Apps mit Spring Boot

Philipp Konopac

Enterprise Applications …

… wird zumindest angerissen



Spring Boot …

… ist das Hauptthema

Bevor wir anfangen …

… und jeder aussteigt, weil nicht klar ist wovon ich rede.

Enterprise Applications	Spring Boot
Groooooooße Anwendung 	Java-Web-Framework mit  Funktionalitäten

Themen …

… und was Euch sonst so erwartet

- Enterprise Applications
  - Definition
  - Grundlagen
- Spring Boot
  - Definition
  - Unterschiede zu Spring
  - Spring Boot Enterprise Application
- Fazit

Zu diesem Vortrag …

… ein paar kleine Infos

- 45 bis 60 Minuten
- Locker und entspannt
- Lieber aufpassen als mitschreiben
- Viele Codebeispiele
  - Vorsicht: Screenshots aus der Seminararbeit sind eventuell nicht aktuell, lieber auf Github schauen
- „YouTube-Tutorial Style“

Ziel …

… wäre zumindest wünschenswert

Verständnis dafür, was Spring Boot ist und warum es sinnvoll ist, es für Enterprise Applications einzusetzen.

Enterprise Applications

Definition

Grundlagen

Enterprise Applications …

… eine kurze Definition

Enterprise Applications sind hoch komplexe Anwendungen, die von Unternehmen genutzt werden. Sie sind oft sehr groß und meistens auch alt, da sie schwer ausgetauscht werden können.

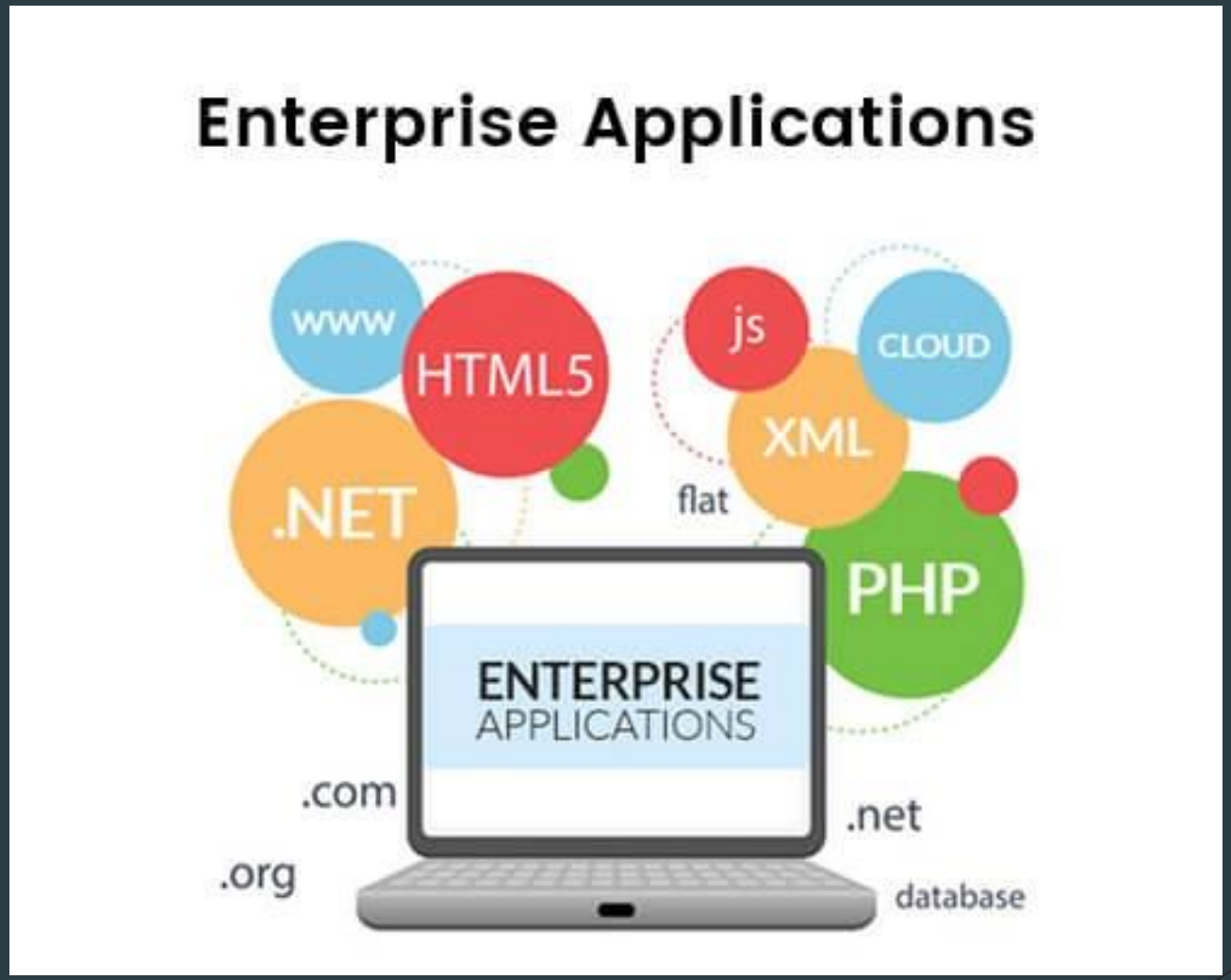
Enterprise Applications …

… und ein paar Grundlagen dazu

- Kommunikation über Netzwerke
- Hohe Sicherheit und Administrierbarkeit nötig
- Oft im Intranet
- Modern: Software-as-a-Service
- Trend: Cloud-Computing
- Typische Anwendungsbereiche:
  - Zahlungsprozess
  - Automatisches E-Mail-Versand-System
  - Abbildung der Geschäftslogik des Unternehmens

Enterprise Applications …

… das Bild von vorhin



Spring Boot

Definition

Unterschiede zu Spring

Spring Boot Enterprise Application

Spring Boot …

… eine kurze Definition

„Ein Webframework […] ist eine Software, die für die Entwicklung von dynamischen Webseiten, Webanwendungen oder Webservices ausgelegt ist. Sich wiederholende Tätigkeiten werden vereinfacht und die Wiederverwendung von Code und die Selbstdokumentation der Software-Entwicklung gefördert. […] Durch vordefinierte und vorgefertigte Klassen werden häufig gebrauchte Funktionen wie Mailversand, sichere Authentifizierung und Authentisierung, Sicherheitsfunktionen, Lokalisierung, Performance (z. B. HTTP Caching) oder grundlegende Funktionen für Webformulare vom Framework mitgebracht.“ (Wikimedia Foundation Inc.)

Spring Boot …

… eine kürzere Definition

- Java-Code für Entwicklung von Webanwendungen
- Vereinfachung von sich wiederholenden Tätigkeiten
- Wiederverwertbarkeit von Code wird erhöht
- Selbstdokumentation durch Namens-Konventionen
- Vordefinierte Klassen für
  - Authentifizierung
  - Performance
  - Lokalisierung
  - …

Spring Boot …

… vs Spring

Spring	Spring Boot
	
Standardmäßig viel Funktionalität	Standardmäßig wenig Funktionalität
Schwergewichtige Anwendung	Leichtgewichtige Anwendung
Viel Konfiguration nötig, auch wenn viele Funktionen nicht gebraucht werden	Wenig bis keine Konfiguration nötig - es wird nur das hinzugefügt und konfiguriert, was auch gebraucht wird

Spring Boot …

… in einer Enterprise Application

- Nötig sind Vorkenntnisse über
  - Java
  - Maven
  - HTTP
  - Datenbanken

Spring Boot …

… Einbindung in mein Projekt

- Erstellung eines gewöhnlichen Maven-Projekts: https://start.spring.io/
- Maven Parent:
  - Group ID: *org.springframework.boot*
  - Artifact ID: *spring-boot-starter-parent*
- Maven Dependency:
  - Group ID: *org.springframework.boot*
  - Artifact ID: *spring-boot-starter-web*
- Github Beispiel

Spring Boot …

… Start-Konfiguration

- *@SpringBootApplication* an die Main-Klasse
- Aufruf *SpringApplication.run(Main.class, args)* in die Main-Methode

Annotation	Funktion
@EnableAutoConfiguration	Automatischer Konfigurationsmechanismus
@ComponentScan	Scannen nach Komponenten
@Configuration	Markiert eine Konfigurations-Klasse
@SpringBootApplication	Dieselbe Funktion wie die oberen drei Annotationen in ihren Standardeinstellungen

- Github Beispiel

Spring Boot …

… Dependency Injection und Beans

- Von Spring verwaltete Klasse = Bean
- Beans implementieren Interfaces (Inversion of Control)
- *@Autowired* zum injecten von Beans
- Beans sind mit Annotationen gekennzeichnet, typische sind
  - *@Component*
  - *@Service*
  - *@Repository*
  - *@Controller*

Spring Boot …

… Komponenten

- *@Component*
- Komponenten werden von Spring gesucht (siehe *@ComponentScan*)
- *@Service*, *@Repository* und *@Controller* werden gefunden, weil diese Annotationen die Annotation *@Component* haben

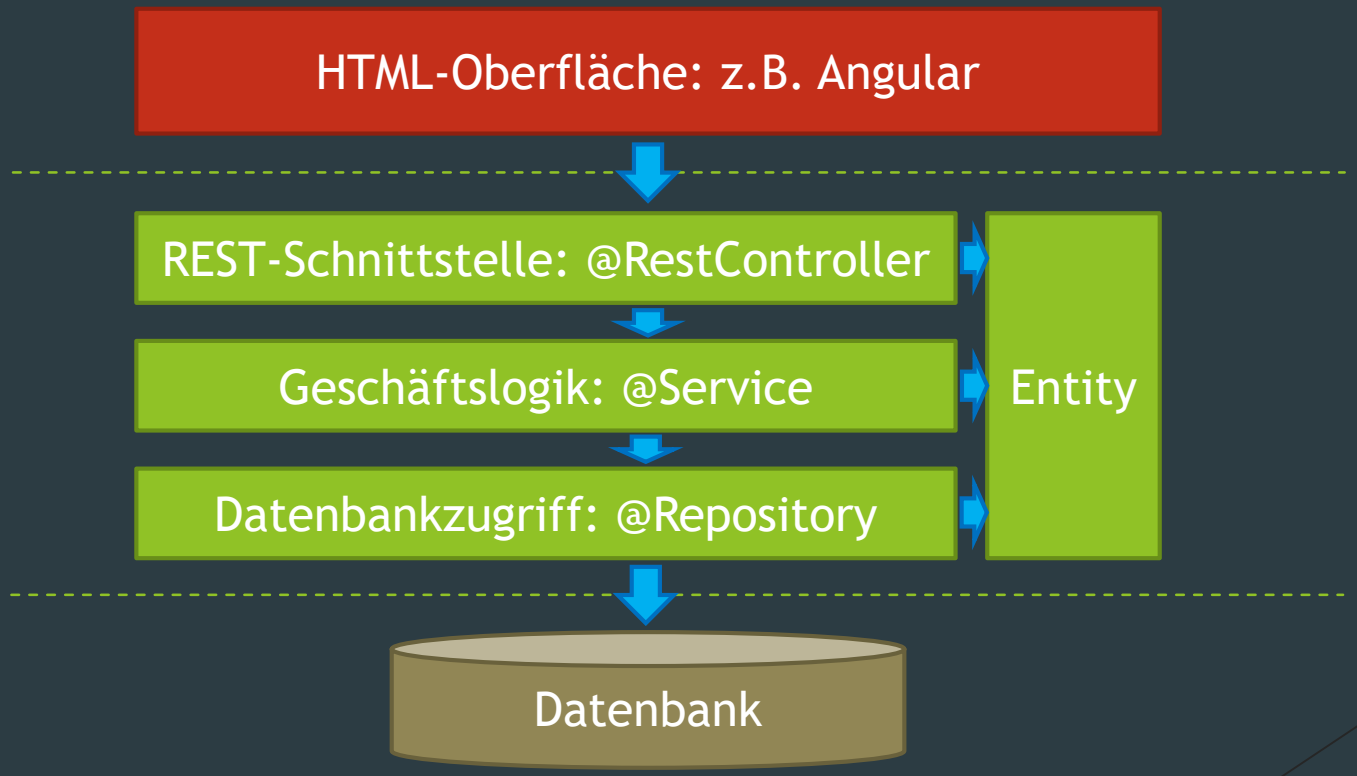
Annotation	Funktion
@Component	Allgemeine Komponente
@Repository	Datenbank-Zugriffs-Klasse
@Service	Geschäftslogik-Klasse
@Controller	Kommunikation mit Außenwelt

- Github Beispiel

Spring Boot …

… Schichtenmodell

- Klassisches MVC-Modell



Spring Boot …

… Persistenzierung

- *@Repository*
- Klasse der Datenbank-Zugriffs-Schicht
- Umwandlung von plattformspezifischen Exceptions in Spring Execptions
- Maven Dependency:
  - Group ID: *org.springframework.boot*
  - Artifact ID: *spring-boot-starter-data-jpa*
- Spring-Data-Klassen Repository und CrudRepository können abgeleitet werden
- CRUD = Create Read Update Delete
- Github Beispiel

Spring Boot …

… Services

- *@Service*
- Geschäftslogik
- Kern der Anwendung
- Keine speziellen Funktionen, außer der Markierung eines Services
  - Könnte sich in Zukunft aber ändern, wer weiß…
- Github Beispiel

Spring Boot …

… REST-Controller

- *@RestController* ist ein spezieller *@Controller*
- Kommunikation mit der Außenwelt
- Mögliche Annotationen an Klassen und Methoden: *@RequestMapping*, *@GetMapping*, *@PostMapping*
- Empfangen von HTTP-Anfragen auf definierte URLs
- Empfangen von Parametern *(@RequestParam*) und Objekten (*@RequestBody*)
- Rücksenden von JSON-Objekten
- Github Beispiel

Spring Boot …

… HTTP-Anfrage

- Protokoll
- Adresse der Anwendung, an die die Anfrage gerichtet wird
- Request Mapping
- Key-Value-Paare für die Request Parameter
  - Bitte nicht wie im Beispiel das Passwort da rein schreiben!

Spring Boot …

… Security und Authentifizierung

- Maven Dependency:
  - Group ID: *org.springframework.boot*
  - Artifact ID: *spring-boot-starter-security*
- Standard-Login Mechanismus aktiviert durch *@EnableWebSecurity*
- Durch Konfiguration können bestimmte URLs ohne Login erreicht werden
- Rechte-Rollen Sytem
- Github Beispiel

Fazit

Fazit …

… was wir gelernt haben

- Spring Boot ist ein Web-Framework das vieles erleichtert
- Konfigurationsaufwand ist schwindend gering
- Lauffähige Anwendung nach nur wenigen Minuten
- Fast alles wird über Annotationen geregelt

Fazit …

… meine Meinung dazu

- Spring Boot macht Spaß!
- Verlockend, aber nicht immer sinnvoll
  - Sollte bei großen Projekten verwendet werden
  - Kleine (Hochschul-)Projekte können dadurch zu kompliziert werden

Danke für Eure Aufmerksamkeit

Noch Fragen?

Quellen …

… mehr davon in meiner Studienarbeit

- https://www.slideshare.net/josebovet/springboot-overview
- *https://www.webopedia.com/TERM/E/enterprise_application.html*
- *https://msdn.microsoft.com/en-us/library/aa267045(v=vs.60).aspx*
- https://www.n-tv.de/wirtschaft/Telekom-fuerchtet-immer-mehr-Cyber-Angriffe-article13892991.html
- https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
- *https://de.wikipedia.org/wiki/Dependency_Injection*
- *https://de.wikipedia.org/wiki/Webframework*
- *https://qph.ec.quoracdn.net/main-qimg-8b974d656ea2a3035f89d731c493180f*
- *https://dzone.com/articles/how-dependency-injection-di-works-in-spring-java-a*