Zürcher Hochschule für Angewandte Wissenschaften

School of Engineering

Bachelor of Science (BSc) in Informatik

Modul Advanced Software Engineering 1 (ASE1)

LE 01 – Spring Framework

Spring Core Framework Basics
Einführung

Institut für Angewandte Informationstechnologie (InIT)

Walter Eich (eicw) / Matthias Bachmann (bacn)

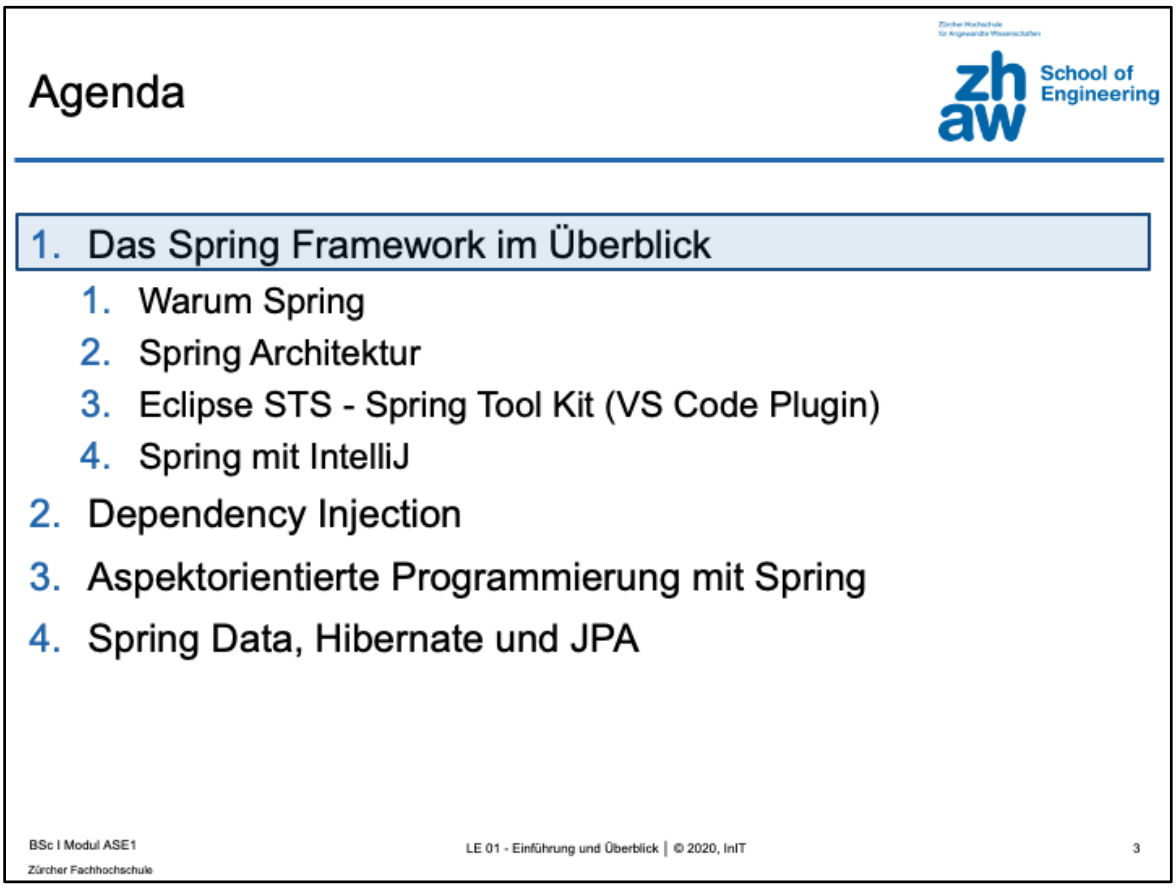
https://www.zhaw.ch/de/engineering/institute-zentren/init/

Das Spring Framework ist wohl die am meisten verbreitete Java-Bibliothek. Der Grund dafür ist relativ einfach, weil es viele häufige Programmieraufgaben auf deutlich weniger komplexe Art und Weise in Angriff nimmt. Binnen weniger Minuten können Java-Entwickler komplexe Anwendungen von Enterprise-Niveau erstellen. Spring kann viel sein, aber auch wenig. Was genau das heissen soll und was Spring Framework konkret bietet, lernen Sie durch dieses Skript. Zunächst sollte man sich vor Augen führen, warum Spring so beliebt in der Community ist.

Lernziele LE 01 – Spring Framework

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering

- Der Studierende,
  - kann die Komponenten des Spring Frameworks erklären
  - kann das Framework und die Entwicklungsumgebung installieren
  - kann Aspekt Orientierte Programmierung anwenden
  - kann JPA mit Spring Repositories einsetzen

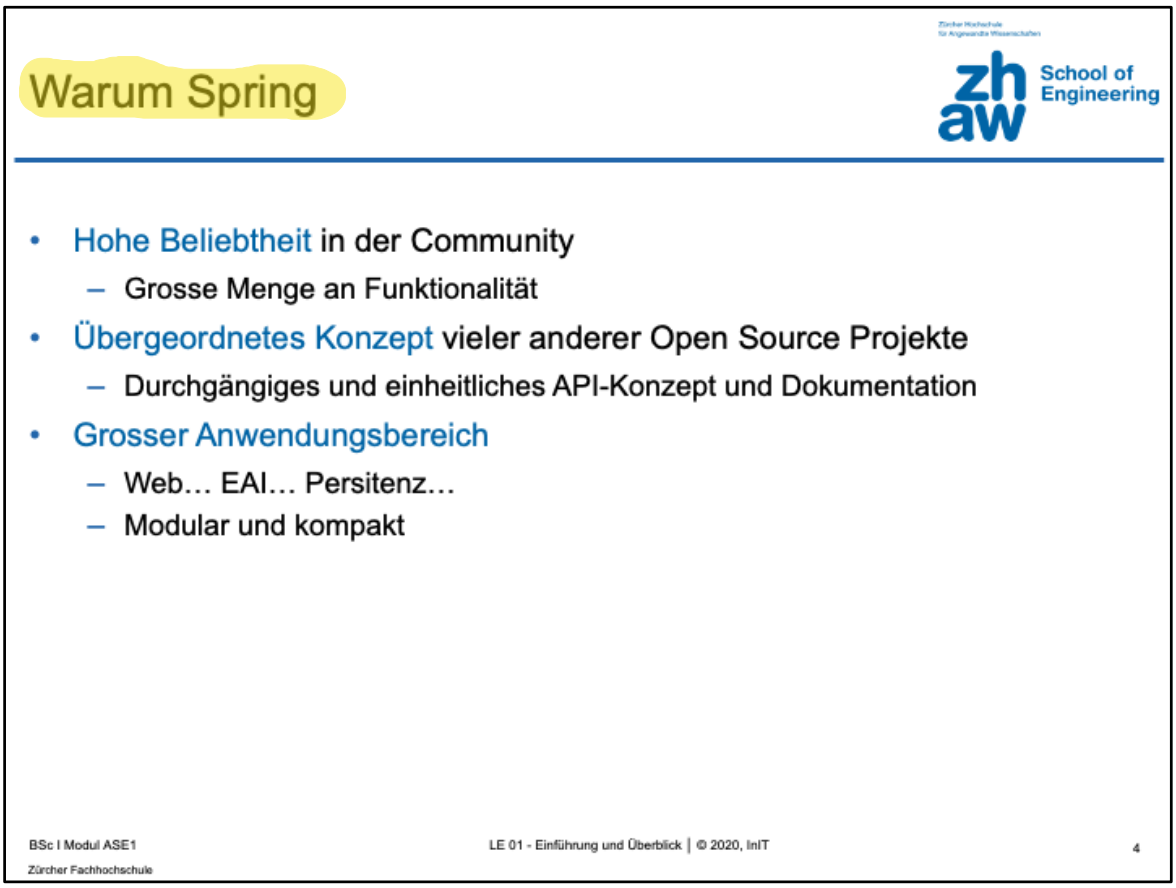


Das Spring Framework im Überblick
Das Spring Framework und all seine zugehörigen Komponenten sind fester Bestandteil des Werkzeugs vieler Java-Entwickler geworden. Durch kontinuierliche Erweiterung bleibt es dauerhaft den Anforderungen moderner Entwicklung gewachsen.

Dependency Injection
Dependency Injection bezeichnet die Art und Weise, wie man die Initialisierung von Abhängigkeiten automatisieren kann und damit Redundanz vermeidet.

Aspektorientierte Programmierung mit Spring
Das Programmierparadigma der AOP erweitert die klassische objektorientierte Programmierung, in der Funktionen der Geschäftslogik in Modulen und Objekte gekapselt werden, um Aspekte. Aspekte drehen sich um die Handhabung von Aufgaben, die über das ganze Software-System hinweg an mehreren Stellen erledigt werden müssen: Logging oder Datenbankzugriffe gehören zu den charakteristischen Bestandteilen.

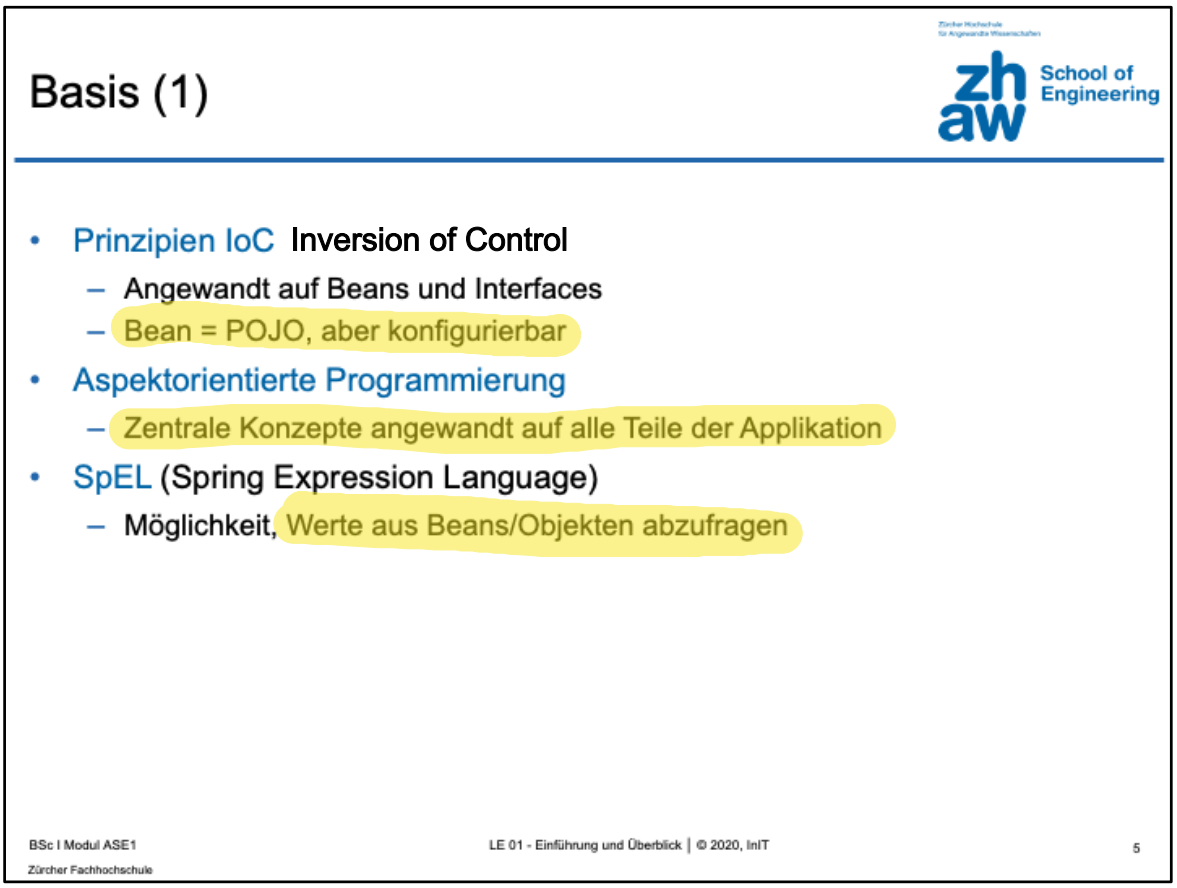
Spring Data, Hibernate und JPA
Mit Spring Data schafft das Framework eine Abstraktion über viele gängige Datenbankzugriffslösungen. Dank des einheitlichen Konzepts ist der Wechsel zwischen verschiedenen Lösungen unkomplizierter denn je.



Hohe Beliebtheit
Neben vielen Anwendungsbereichen, die direkt in Spring implementiert sind bietet Spring auch eine Art Dachverband vieler anderer Open-Source-Projekte.

Übergeordnetes Konzept
Dachverband heisst, es nutzt andere Open-Source-Projekte und bietet ein einheitliches API-Interface um dementsprechend mit diesen Projekten arbeiten zu können. Dadurch sind auch viele der Open-Source-Projekte, die darunter liegen, wie beispielsweise Hibernate einfach austauschbar ohne die eigene Anwendung komplett verändern zu müssen.

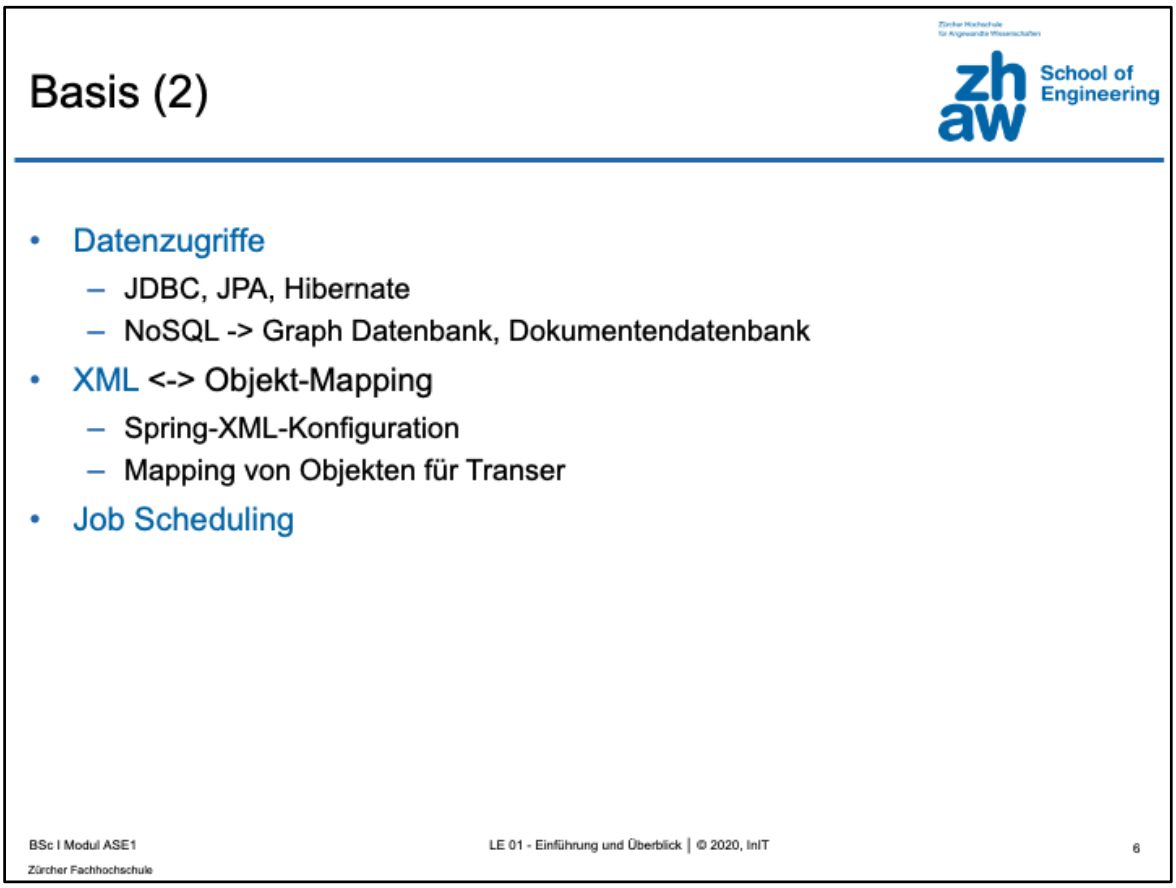
Grosser Anwendungsbereich
Ausserdem kümmert sich das Spring Framework um einen sehr grossen Anwendungsbereich und vor allem auch um die Schnittstellen zwischen diesen Anwendungsbereichen. Das betrifft sowohl Web- als auch Enterprise-Architekturintegration als auch Persistenz und viele mehr. Trotz der vielen Möglichkeiten die Spring bietet, ist es immer noch modular und kompakt, d. h. man nutzt immer nur das, was man tatsächlich braucht. Neben all den verschiedenen Dingen, die das Spring Framework bietet, gibt es auch einige Kernkomponenten, die unabdingbar sind, damit das Spring Framework funktionieren kann.



Prinzipien IoC
Unter anderem die Prinzipien von Inversion of Control, anders genannt auch Dependency Injection. Diese werden auf Beans und Interfaces angewendet um automatische Kopplung zu garantieren. Eine Bean als solches ist nichts weiter als ein klassisches Java-Objekt das konfigurierbar ist, manchmal auch POJO genannt, Plain Old Java Object.

Aspektorientierte Programmierung
Neben den Prinzipien der Dependency Injection ist auch die aspekt-orientierte Programmierung äusserst wichtig. Sie sorgt dafür, dass das Spring Framework zentrale Konzepte implementieren kann, die sich auf alle Teile der Applikation selbst auswirken können.

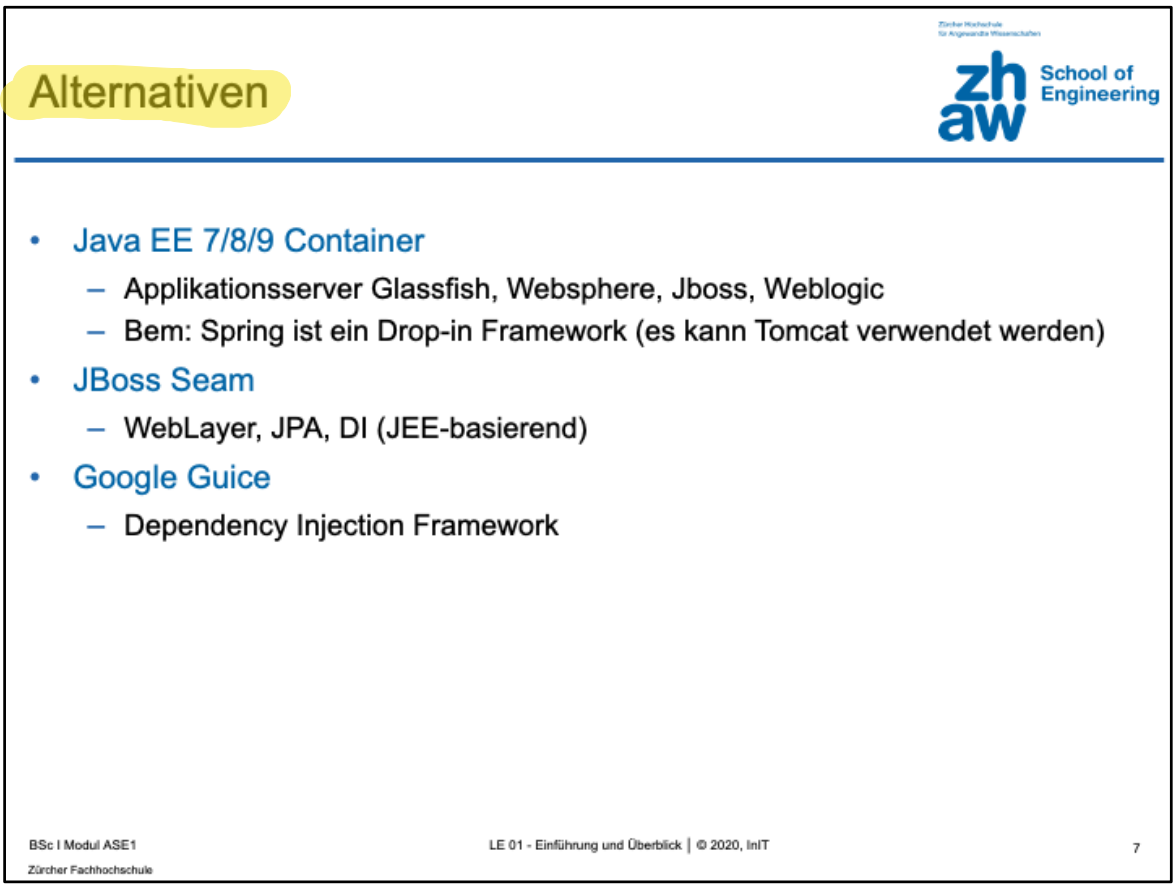
SpEL (Spring Expression Language)
Weiterhin gibt es noch die Spring Expression Language, die ähnlich den Konzepten von JSP – *Java Server Pages* es möglich macht, Objekte, bzw. *Beans* nach bestimmten Attributen abzufragen.



Datenzugriffe
Neben diesen Kernfunktionalitäten gibt es auch noch einige andere Aspekte, die äusserst wichtig sind. Zum Beispiel wird der Zugriff auf Daten harmonisiert. Das bedeutet, es gibt eine Austauschbarkeit zwischen Konzepten wie JPA, Hibernate und es gibt auch neue Möglichkeiten JDBC zu nutzen. Abgesehen von den klassisch relationalen Datenbankzugriffskonzepten gibt es auch schon die Möglichkeit mit NoSQL-Datenbanken zu interagieren, wie Grafendatenbanken, Dokumentendatenbanken und vielen anderen.

XML
Spring zeichnet sich vor allem auch durch seine Konfigurierbarkeit mit XML aus denn Spring *mappt* unkompliziert zwischen XML-Repräsentationen eines Objektes und dem Objekt selbst. Auch neben der XML-Konfiguration kann das für viele andere Möglichkeiten genutzt werden, wie die Übertragung von Objekten über Webservices.

Job Scheduling
Auch in den Bereichen Scheduling und Workflows bietet Spring viele Möglichkeiten. Dadurch ersetzt es zusätzliche Bibliotheken wie Quartz oder dem Arbeiten mit umständlichen Zeitobjekten. Spring ist allerdings nicht allein in seinem Anwendungsbereich.



Java EE 7 Container

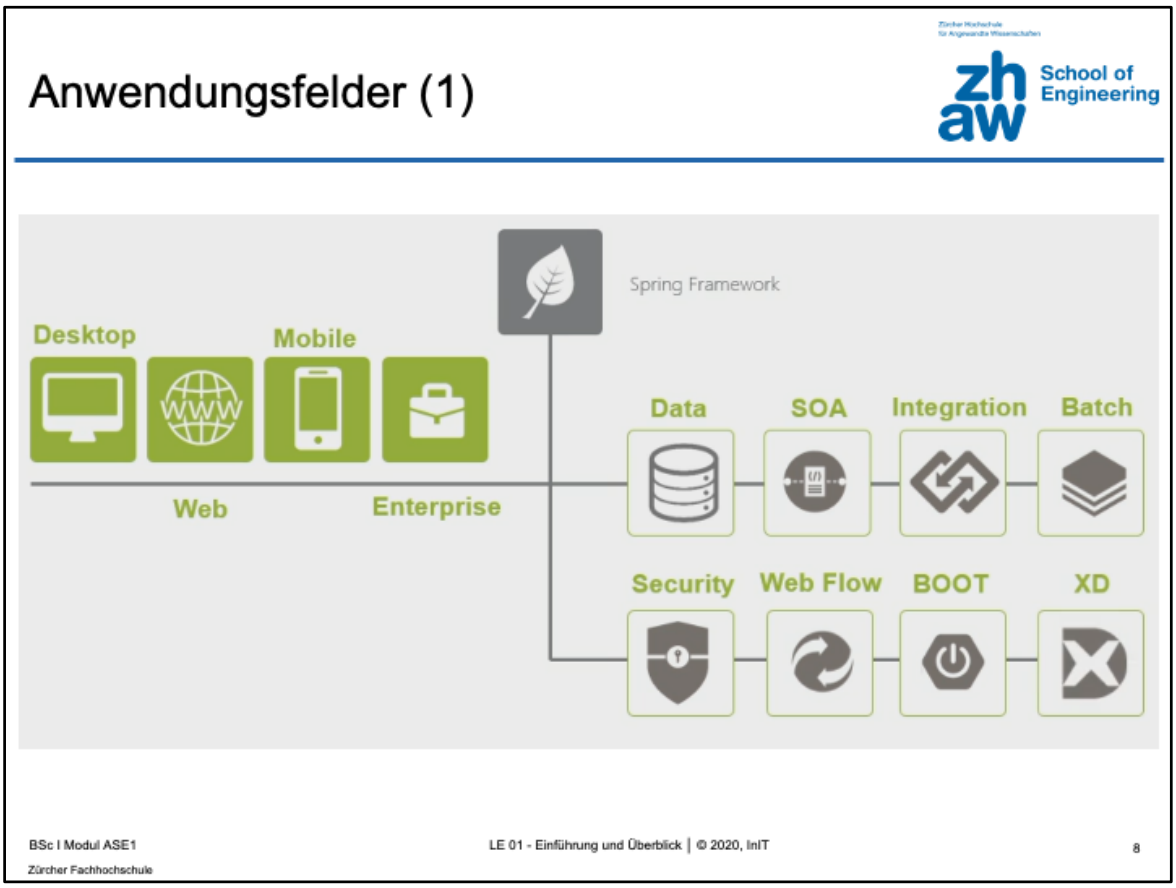
Es gibt auch gängige Alternativen zu Spring. Eine der wichtigsten ist die *Java Enterprise Edition*. Viele der Konzepte, die in Spring verwendet werden sind einerseits kompatibel mit der *Java Enterprise Edition*, basieren auf dieser, weshalb auch die Austauschbarkeit zwischen Spring und Java Enterprise Edition unter bestimmten Umständen gegeben sein kann. Grundsätzlich bemühen sich beide um die gleichen Konzepte. Anders als bei der *Java Enterprise Edition* ist *Spring* ein *Drop-In* Framework, d. h. der Applikationsserver kann beispielsweise ein *Tomcat* sein, der basistechnisch keine Funktionalität unterstützt. Während ein Server wie *Glassfish*, Websphere oder JBoss schon einige Funktionalitäten mit sich bringt, die auf die Java Enterprise Edition zugeschnitten sind.

JBoss Seam Framework

Als Drop-In Framework gibt es auch alternativ das sogenannte *JBoss-Seam-Framework*. Dieses implementiert auch viele Dinge regulär z. B. *Java Enterprise Standard*.

Google Guice

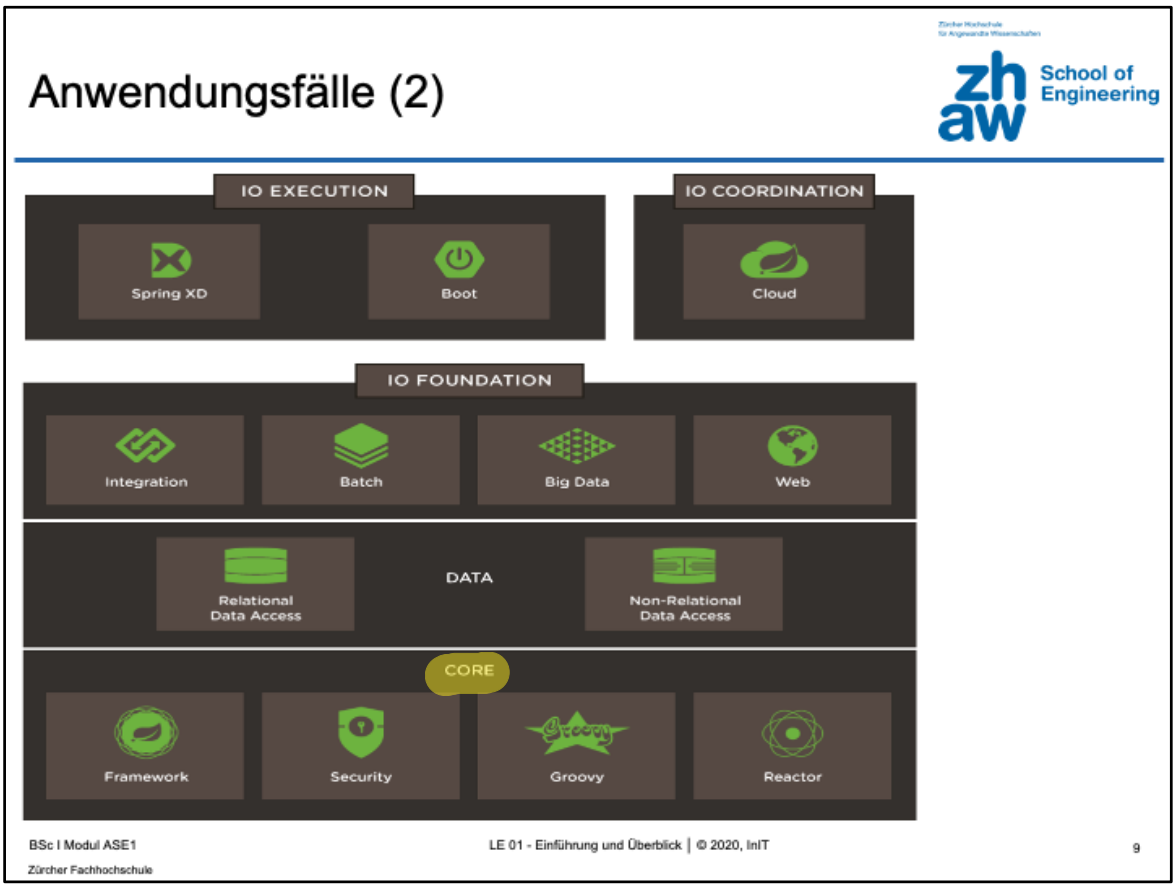
Die Kernfunktionalität von Spring Framework wie die *Dependency Injection* wird auch von *Google Guice* abgedeckt, denn *Google Guice* ist die Referenz-Implementierung für den Java Standard was *Dependency Injection* angeht.



Anwendungsfelder
Die Anwendungsfelder des *Spring-Framework* sind vielfältig und nicht immer nur auf Enterprise-Anwendungen beschränkt. Grundsätzlich eignet sich das Spring Framework für die Entwicklung von Desktopapplikationen, von Webanwendungen im speziellen, von mobilen Anwendungen mit Android, das ohnehin auch auf Java basiert, als auch für konkret Enterprise- Anwendungen. Dafür bietet Spring eine Reihe verschiedener Komponenten, unter anderem Spring Data, das sich eben mit verschiedenen Arten von Datenbanken auseinandersetzt und objektrelationalem Mapping.

Die Möglichkeit serviceorientierte Architekturen zu bauen, z. B. mit Webservices via Rest. Die Möglichkeit Integrationsschnittstellen zwischen verschiedenen Anwendungen zu bauen als auch grosse Mengen von Daten mit Batch-Processing zu bearbeiten. Weiterhin bietet es auch die Möglichkeit Security-Aspekte zu implementieren und somit eine anwendungsweite Sicherheit zu garantieren. Für Webanwendungen schafft es mit *Webflow* die Möglichkeit, Nutzer durch eine bestimmte Anzahl von Seiten zu geleiten. *Spring-Boot* ist die hauseigene Implementierung, die auf einem *Tomcat, Jetty oder Undertow Webserver* aufbaut und somit eine Art Drop-In bietet für Spring-Anwendungen.

Eins der jüngsten Projekte, *Spring XD*, setzt sich dann mit den neuesten Konzepten rund um Big Data auseinander.



Links
Übersicht auf die verschiedenen domänenspezifischen Projekte:
http://spring.io/docs/reference

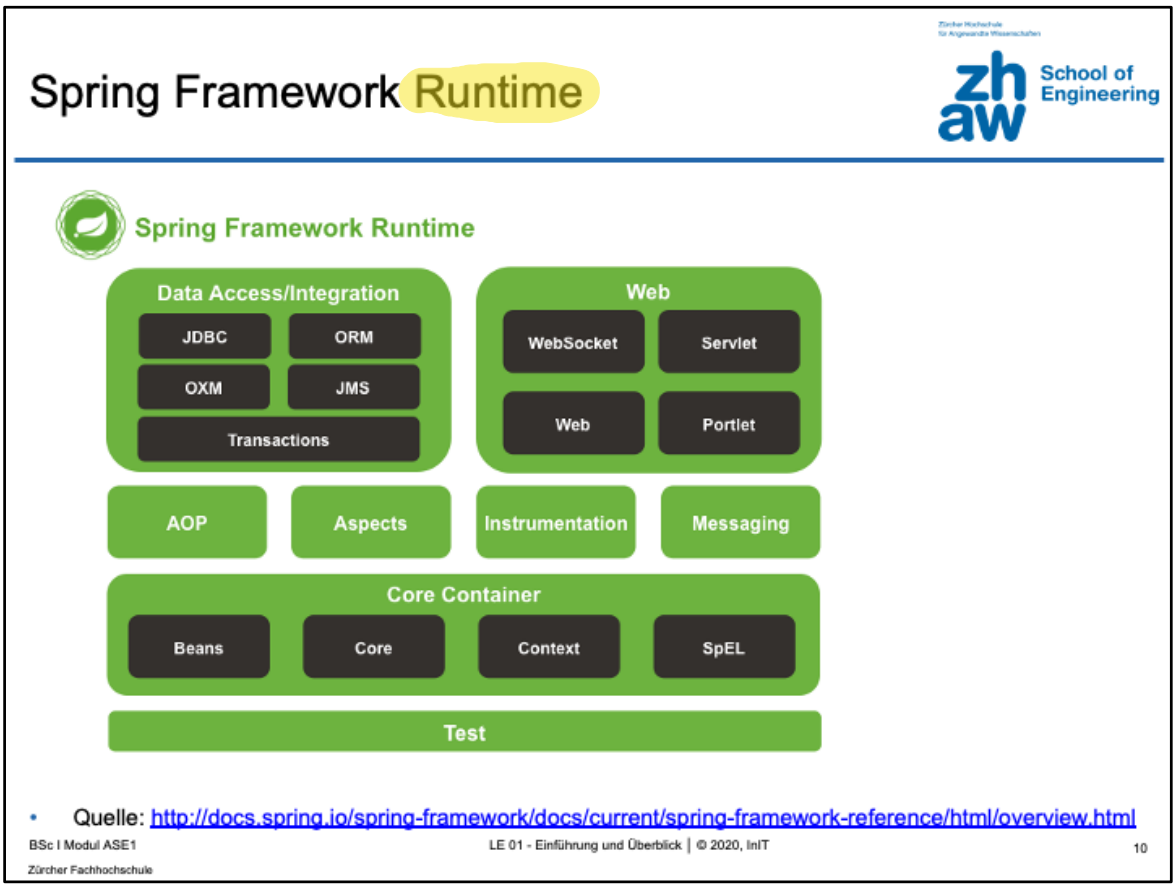
Spring Framework Dokumentation:
http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.htm

Spring Boot Dokumentation
http://docs.spring.io/spring-boot/docs/
http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#getting-started-installation-instructions-for-java

Initializr
https://start.spring.io/

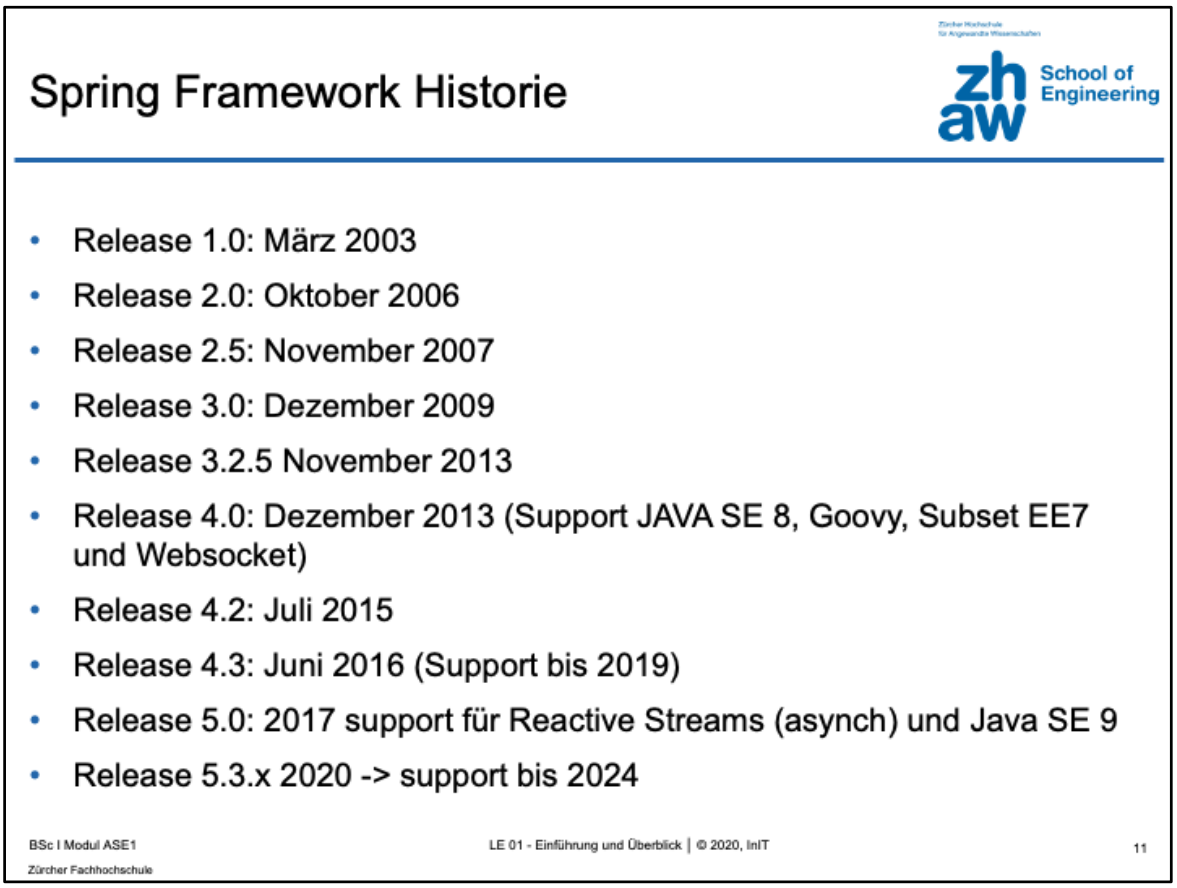
STS - Spring Tools
https://spring.io/tools

Security
http://projects.spring.io/spring-security



Core-Container
Der Kern des ganzen Frameworks lässt sich gut in einer Art Architektur darstellen, mit verschiedenen Ebenen. Die Basis aller Dinge ist das sogenannte Core-Framework oder auch der Core-Container. Dort werden die Möglichkeiten von Beans abgebildet, die Möglichkeiten der Dependency Injection und des Kontextes. Aufbauend auf diesen Konzepten kann die aspektorientierte Programmierung, die Aspekte konkret, die Instrumentalisierung, die dabei stattfindet und der Messaging-Austausch zwischen den Komponenten stattfinden.

Data Access/Integration und Web
Diese ermöglichen die Funktionalitäten auf den obersten Ebenen, z. B. in der Datenpersistenz. Dort deckt Spring beispielsweise Transaktionen über verschiedene Datenbanktechnologien und Implementierungen der Interfaces hinweg ab. Neben der Datenpersistenz ist auch der Web-Bereich besonders wichtig der hier auf dem Konzept von Servlets aufbaut. Unter all diesen Komponenten steht die Möglichkeit, einheitlich über alle Dinge hinweg testen zu können. Genau dieses einheitliche Testkonzept erlaubt es auch, relativ einfach über alle Ebenen hinweg Tests schreiben zu können und dementsprechend auch zu garantieren, dass die Anwendung voll funktional ist.



Release 1.0 und 2.0
In seinem im **Oktober 2002** erschienen Buch "*Expert One-on-One J2EE Design and Development*" stellt *Rod Johnson* die Grundlagen und Prinzipien eines alternativen *J2EE -Frameworks* vor. Ein Grundgerüst des Frameworks ist dem Buch bereits beigelegt. Das Interesse ist so gross, dass das Framework als OpenSource Projekt weiterentwickelt wird und daraufhin im Juni 2003 die Version 1.0 von Spring erscheinen kann. Im Juni 2006 erscheint dann die Version 2.0, Spring gewinnt in diesem Jahr den "JAX Innovation Award" und ist bis zu diesem Zeitpunkt bereits mehr als 1 Million mal aus dem Internet geladen worden.

Release 2.5
Am 19. November 2007 wurde die Version 2.5 veröffentlicht. Ursprünglich war die Entwicklung als Version 2.1 vorgesehen, aber auf Grund der vielen neuen Funktionen wurde stattdessen die Version 2.5 Nachfolger für die Version 2.0.x. Spring 2.5 unterstützt vollständig die Java-6-Version sowie die *Java-EE in der Version 5* bei gleichzeitiger Abwärtskompatibilität zu Java 1.4 sowie Java EE 1.3. Ausserdem werden in Spring 2.5 Annotations für die Konfiguration der Anwendungskomponenten unterstützt.

Release 3.0
Im September 2009 wurde SpringSource von VMWare übernommen. Version 3.0 erschien am 16. Dezember 2009. Neu integriert wurden darin unter anderem eine Expression Language u. a. für die Konfiguration der Spring Beans und eine Unterstützung für REST. Ausserdem wurde die Java-basierte Konfiguration aus dem Spring Java Config-Projekt in Spring 3.0 integriert.

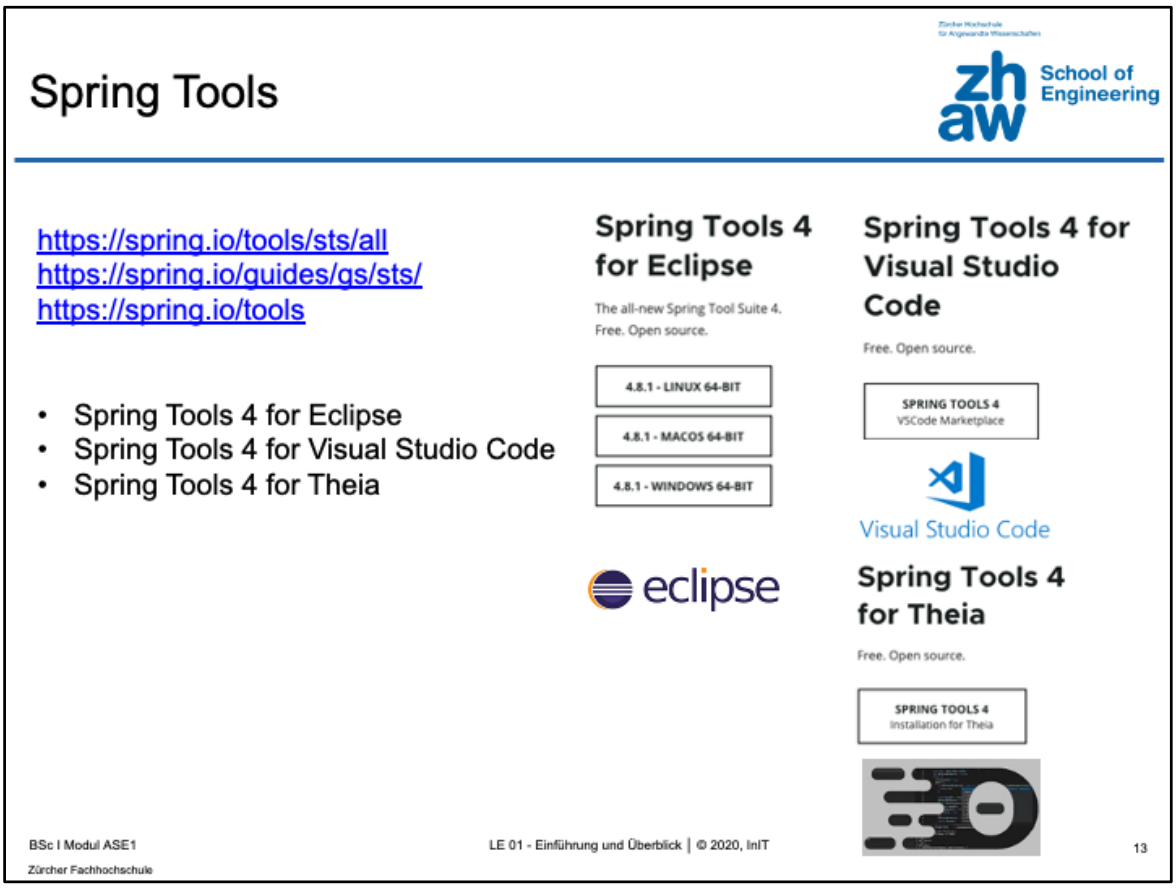
Release 3.1.
Wesentliche Neuerungen der Version 3.1 waren die portable Cache-Abstraktion und die Unterstützung von Konfigurationsprofilen, bei denen abhängig von der Umgebung unterschiedliche Spring-Beans erzeugt werden. Ausserdem werden *Web-Conversations* eingeführt, mit denen Zustand verwaltet werden kann, der über mehrere Web-Seiten zur Verfügung stehen muss.

Aktuell ist die Version 4.x.x
Das Spring-Framework wird nach wie vor durch die Spring-Kernentwickler, unter anderem Jürgen Höller, Rod Johnson und Rob Harrop, weiterentwickelt. VMWare und EMC Corporation haben ein JointVenture mit dem Namen **Pivotal** gegründet.

JDK Bereich

- Spring Framework 5.3.x: JDK 8-17 (expected)
- Spring Framework 5.2.x: JDK 8-15
- Spring Framework 5.1.x: JDK 8-12
- Spring Framework 5.0.x: JDK 8-10
- Spring Framework 4.3.x: JDK 6-8

- JDK LTS Versionen: JDK8, 11, 17 (2021)



Die Spring ToolSuite ist die Entwicklungsumgebung, die das Spring-Projekt selbst bereitstellt. Sie unterstützt den Entwicklungsprozess an vielen Stellen und nimmt viele der Komplexitäten rund um Konfiguration und Abhängigkeitsverwaltung in Angriff.

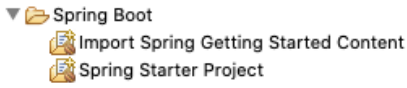
Die komfortabelste Art zu entwickeln ist mit einer IDE, die möglichst viel kontextsensitive Unterstützung bietet. Besonders im Zusammenhang mit der Konfiguration von Spring-XML-Dateien sind genau diese Unterstützungsmöglichkeiten von grossem Vorteil. Neben den vielen anderen IDEs, die sich darum bemühen, möglichst guten Support zu liefern, ist die Spring Tool Suite, die vom Spring-Projekt selbst stammt, eines der komfortabelsten Produkte, wenn man mit Spring arbeiten möchte.

Variante 1: STS Standalone
Variante 2: STS in existierendes Eclipse

Eclipse – New Projects

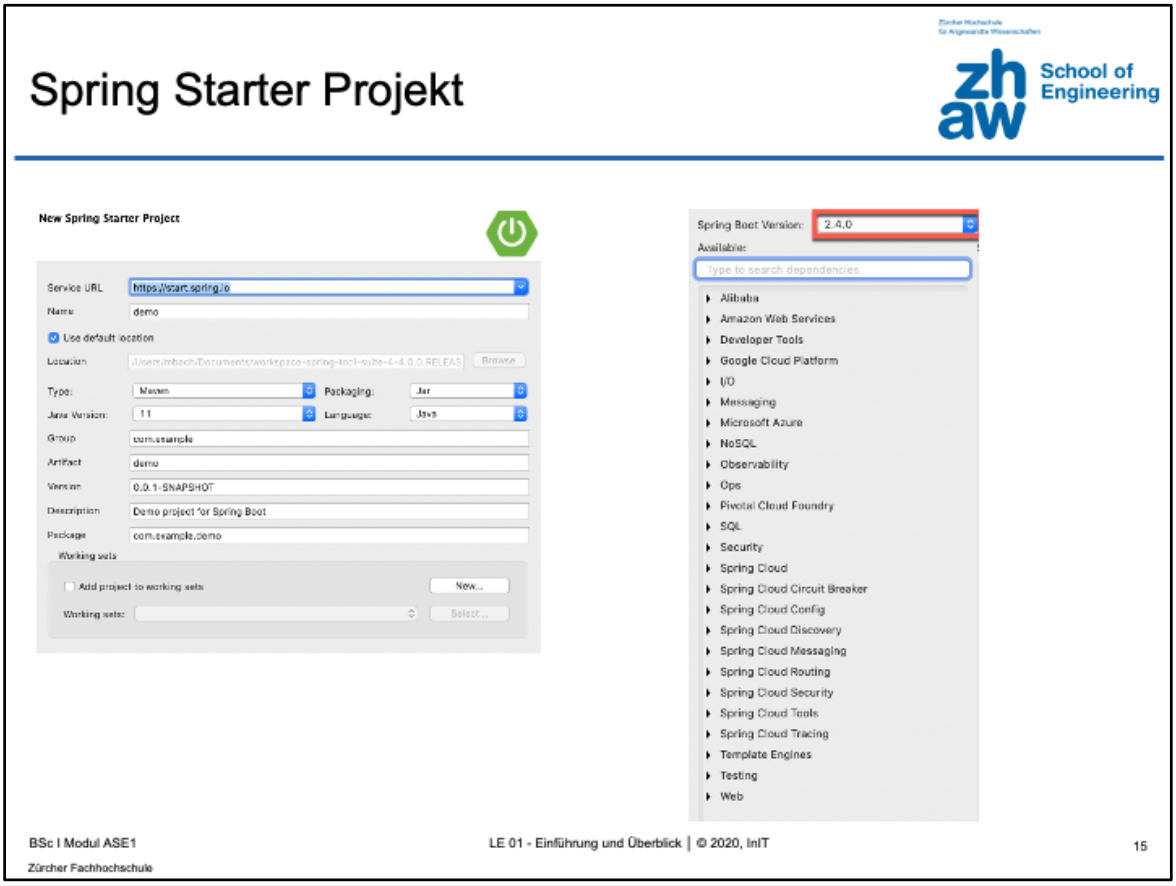
School of Engineering

- File -> New -> Other



Wenn man auf der grünen Wiese anfangen möchte, eignen sich am besten immer die Maven- oder Gradle-Projekte. Einerseits die reguläre Spring Maven-Applikation und andererseits das Spring Web-Projekt. Das Projekt wurde erstellt und kann direkt genutzt werden. In diesem vorkonfiguriertem Projekt befinden sich in der Maven pom.xml schon die entsprechenden Einträge für Abhängigkeiten zu Spring. Diese kann man über *Dependencies* einsehen, und sieht hier wie die Abhängigkeiten *spring-context*, *spring-transactions* und auch andere Dinge um Spring eingebunden sind.

Die Legacy Projekte beinhalten die Konfiguration von Spring mittels XML-Dateien. Meistens wird jedoch *Spring-Boot* verwendet um die Konfiguration mittels Annotationen vorzunehmen.



Spring Starter Projekt

Neben den Projekten, die von der grünen Wiese starten, kann man mit der Spring Tool Suite auch einige der *Spring-Starter-Projekte* beginnen. Diese ermöglichen es, Komponenten welche man für sein Projekt genau braucht, auszuwählen. Sollte man sich für einige praktische Beispiele interessieren, gäbe es auch die Möglichkeit über Import *Spring-Getting-Started-Content* vordefinierte Projekte des Spring Frameworks zu wählen. Eine der beliebtesten dieser Anwendungen ist z. B. die *spring-petclinic*, die eine komplette Spring-MVC-Anwendung zeigt.

Maven / Gradle Dependencies (1)

School of Engineering

GroupId	ArtifactId	Description
org.springframework	spring-aop	Proxy-based AOP support
org.springframework	spring-aspects	AspectJ based aspects
org.springframework	spring-beans	Beans support, including Groovy
org.springframework	spring-context	Application context runtime, including scheduling and remoting abstractions
org.springframework	spring-context-support	Support classes for integrating common third-party libraries into a Spring application context
org.springframework	spring-core	Core utilities, used by many other Spring modules
org.springframework	spring-expression	Spring Expression Language (SpEL)
org.springframework	spring-instrument	Instrumentation agent for JVM bootstrapping
org.springframework	spring-instrument-tomcat	Instrumentation agent for Tomcat
org.springframework	spring-jdbc	JDBC support package, including DataSource setup and JDBC access support

Maven:

```
<dependencies>
   <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>4.3.2.RELEASE</version>
      <scope>runtime</scope>
   </dependency>
</dependencies>
```

Gradle:

```
dependencies { compile("org.springframework:spring-context:4.3.2.RELEASE")
testCompile("org.springframework:spring-test:4.3.2.RELEASE") }
```

In ASE1 extra Spring verwenden und selber konfigurieren, anstatt Spring Bott.
In ASE2 Spring Boot lernen und verstehen wie gut es ist mit der Konfiguration.

Maven / Gradle Dependencies (2)

School of Engineering

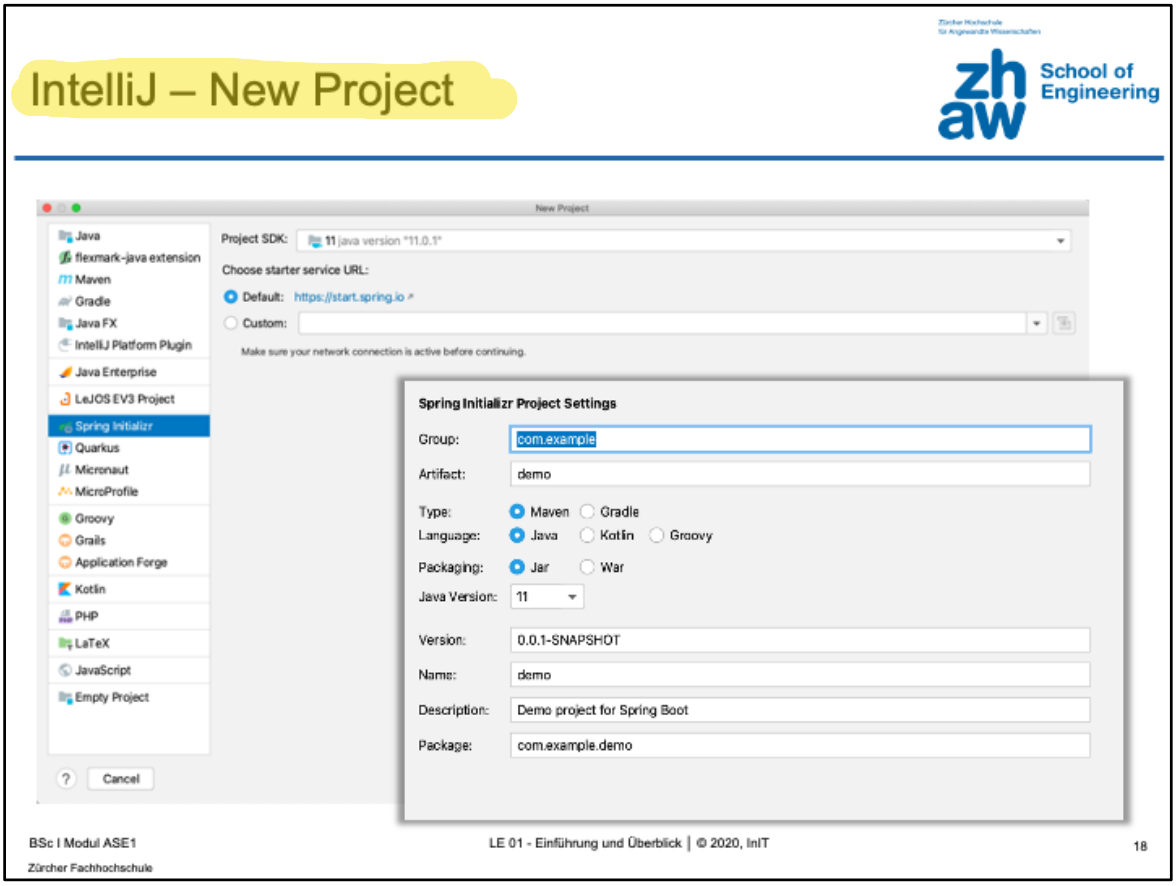
GroupId	ArtifactId	Description
org.springframework	spring-jms	JMS support package, including helper classes to send and receive JMS messages
org.springframework	spring-messaging	Support for messaging architectures and protocols
org.springframework	spring-orm	Object/Relational Mapping, including JPA and Hibernate support
org.springframework	spring-oxm	Object/XML Mapping
org.springframework	spring-test	Support for unit testing and integration testing Spring components
org.springframework	spring-tx	Transaction infrastructure, including DAO support and JCA integration
org.springframework	spring-web	Web support packages, including client and web remoting
org.springframework	spring-webmvc	REST Web Services and model-view-controller implementation for web applications
org.springframework	spring-webmvc-portlet	MVC implementation to be used in a Portlet environment
org.springframework	spring-websocket	WebSocket and SockJS implementations, including STOMP support

Maven Beispiel mit log4j:

```
<dependencies>
   <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>4.3.2.RELEASE</version>
   </dependency>
   <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>1.2.14</version>
   </dependency>
</dependencies>
```

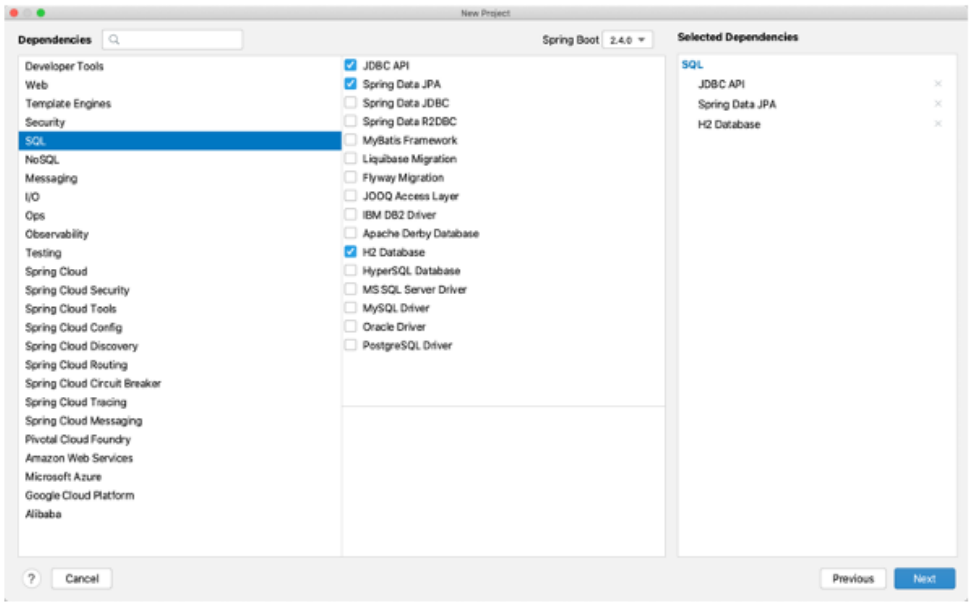
Beispiel log4j.properties Datei um auf die Konsole zu loggen:

```
log4j.rootCategory=INFO, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %t %c{2}:%L - %m%n
log4j.category.org.springframework.beans.factory=DEBUG
```



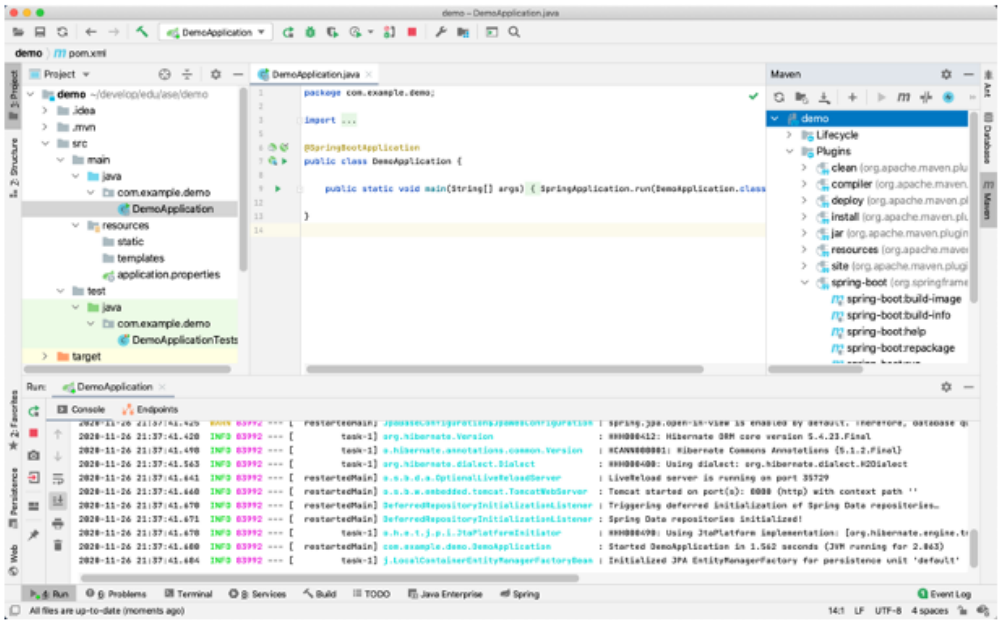
Neues Projekt erstellen
Wenn man ein Projekt auswählt, , z. B. *Spring-Data-JPA*, auf *Next* klickt, kann man das Projekt erstellen. Für manche Arten von Projekten gibt es auch Beispiele, wie z. B. *Spring-Batch.* Dort hat man die Möglichkeit das Projekt ausgehend von einem Template zu erstellen. Man gibt dafür einen Namen ein und bestätigt das Ganze mit *Finish*. Das Projekt wird erstellt und die Abhängigkeiten aufgelöst. Sollte man ein reines Java-Projekt erstellen, muss man sicherstellen, dass über den Rechtsklick bei Add-Framework-Support gegebenenfalls Spring nachgezogen wird.

IntelliJ – New Project (Initializr)



Mittels *Spring-Initializr* können Sie entscheiden, welche Abhängigkeiten Sie dem Projekt hinzufügen möchten. Die Grundlage für diese Auswahl ist Spring Boot (aktuell in der Version 1.4.x).

Spring Demo Code Generation



Für die Entscheidung zwischen der Lösung *Spring-Tool-Suite* oder *IntelliJ-Idea* macht man sich am besten mit beiden Tools einmal vertraut und sucht sich das Tool aus, das den eigenen Arbeitsfluss am besten unterstützt. Aus der persönlichen Erfahrung heraus, ist es einfacher mit der *Spring-Tool-Suite* zu starten, da dort einige Beispiele dabei sind. Für den täglichen Programmfluss ist es deutlich einfacher und schneller mit *IntelliJ-Idea* zu arbeiten, da es wesentlich weniger komplex ist und die meisten Aufgaben deutlich schneller bewältigen kann.

Zusammenfassung Einführung

- Spring Basis
- Spring Anwendungsfälle
- Spring Framework Architektur
- Historie
- Spring Tools Suite auf der Basis von Eclipse
- Spring Starter und Initializr
- Spring mit IntelliJ