Spring Framework 6

Beginner to Guru

Introduction to Spring Framework

About Spring Framework

- Spring Framework is the most popular Java framework for building enterprise grade applications

- Enterprise Grade being highly scalable and reliable applications

  - Over 60% of companies using Java use Spring

- Highly popular with banks, financial institutions, and large retailers

- Can be used for traditional monolithic applications

- Well suited for modern micro service based architectures

- Commonly used as the "backend" technology

Introduction to Spring Framework

- Introduced by Rod Johnson in 2003

  - Started as a simpler alternative to J2EE

  - Rod Authored book called "Expert One on One J2EE without EJB"

  - EJB - Enterprise Java Beans, aka "XML Hell"

  - Focus was on using POJOs to simplify development

- March 2004 Spring Framework 1.0 Released

  - Quickly adopted in the Java community

Spring Framework History

- 2007 Spring Framework 2.5 Released with Annotation based configuration

- August 2009 Spring Source purchased by VMWare for $420 Million

- December 2009 Spring 3.0 Released with Java based configuration

- April 2013 - VMware forms new company, Pivotal. All Spring applications moved to this company

- April 2014 - Spring Boot 1.0 Released

- July 2017 - Spring Framework 5 Released, Introduction of Reactive Programming

- Nov 2017 - Spring Boot 2.0 Released

Spring Framework History

- Fall 2022 - Spring Framework 6.0 Released

  - Requires Java 17+

  - Spring Framework 5.x supports Java 8 - Java 17
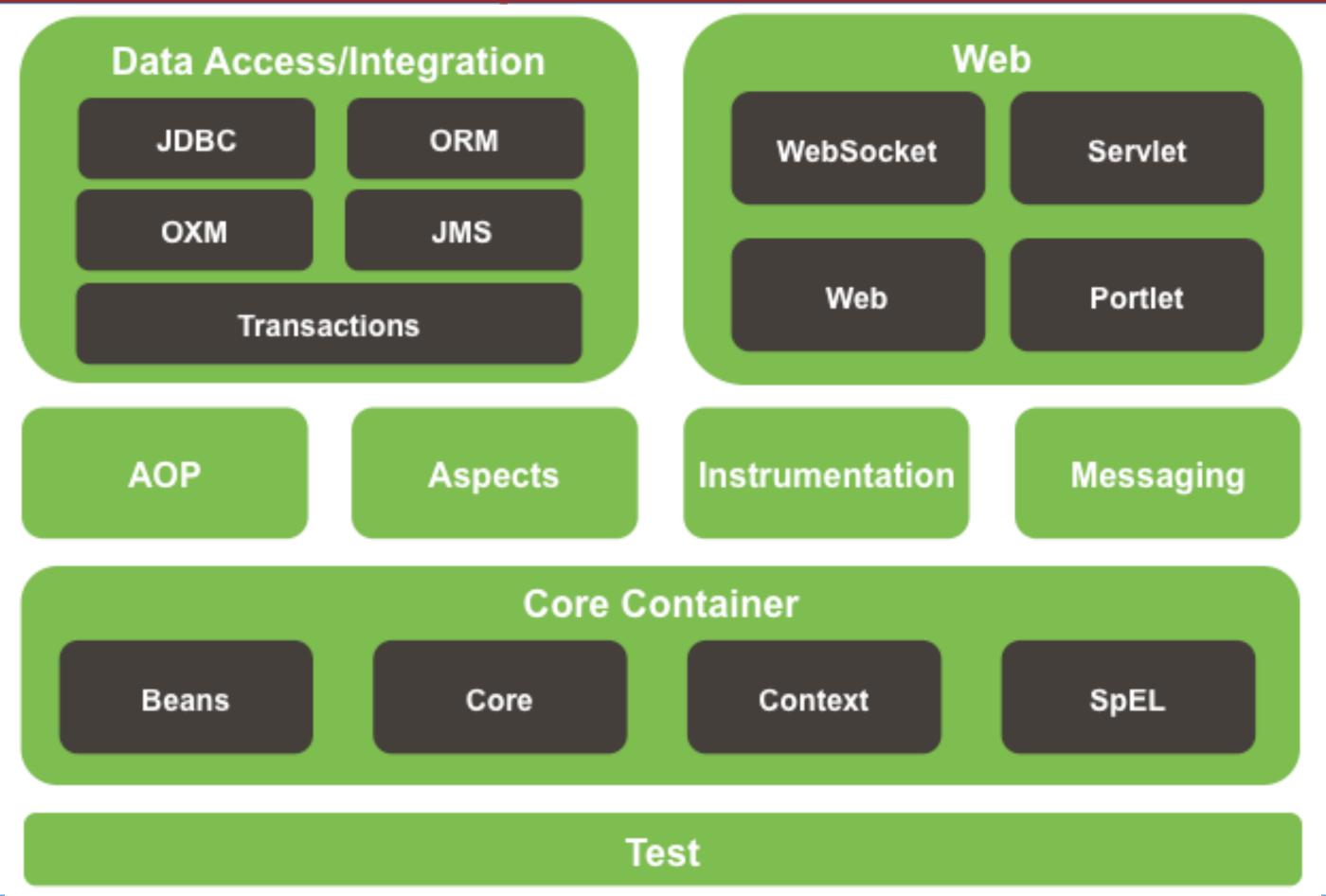
- Fall 2022 - Spring Boot 3.0 Released

Spring Framework vs Spring Boot

- **Spring Framework** is a collection of framework libraries

  - Dependency Injection, Web, Transaction Management, etc

- **Spring Boot** is automated tooling for Spring Framework applications

  - Think of it as a wrapper around Spring

  - You can use Spring Framework without Spring Boot

  - But you cannot use Spring Boot without Spring Framework

Spring Boot Features

- Curated "Starter" Dependencies for common components

- Sensible "Auto-Configuration" based on classes found on the classpath

  - For example, will auto-configure an in memory database if H2 is on the class path

- Externalized Configuration via files and environment variables

- Logging auto-configuration

- Performance Metrics

- Healthcheck endpoints

- Enhanced failure information

Spring Projects

- **Spring Data** - Collection of projects for persisting data to SQL and NoSQL databases

- **Spring Cloud** - Tools for distributed systems

- **Spring Security** - Authentication and Authorization

- **Spring Session** - Distributed web application sessions

- **Spring Integration** - Enterprise Integration Patterns

- **Spring Batch** - Batch processing

- **Spring State Machine** - Open Source State Machine

SPRING FRAMEWORK
GURU