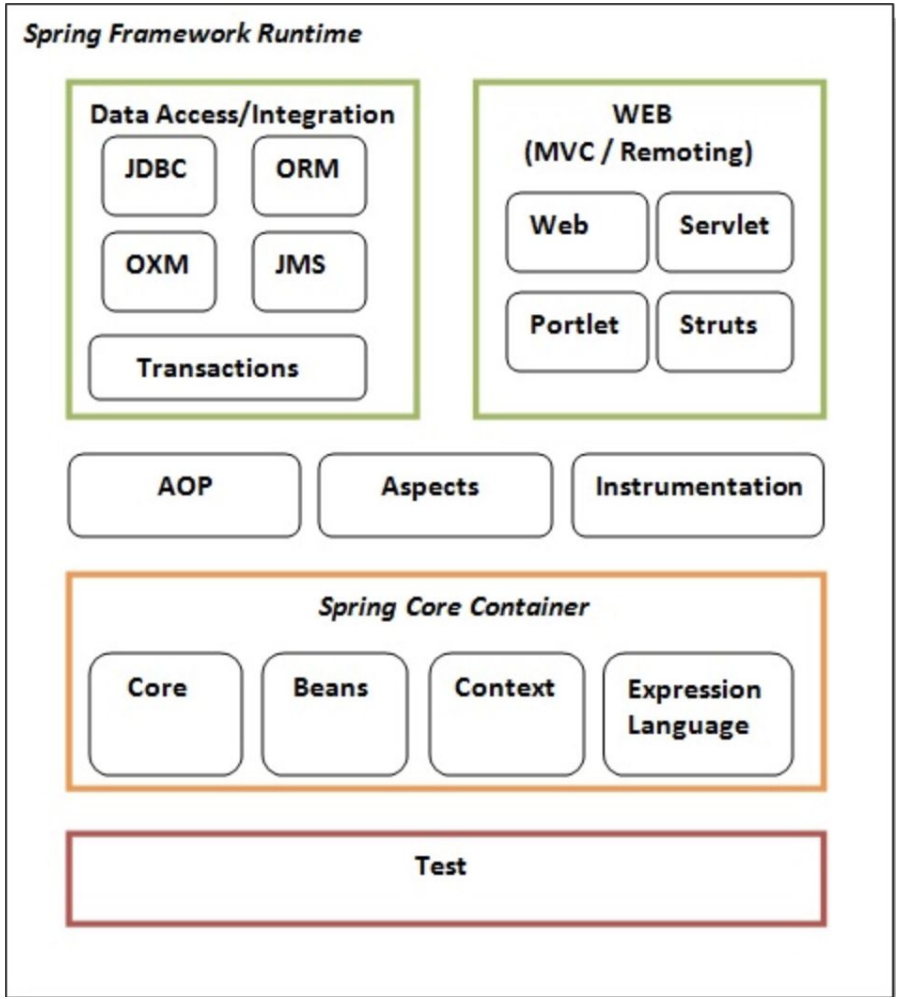
Spring framework

What is spring and why use it?

- Spring is the most popular application development framework for enterprise Java. Millions of developers around the world use Spring Framework to create high performing, easily testable, and reusable code.
- Spring framework is an open source Java platform. It was initially written by Rod Johnson and was first released under the Apache 2.0 license in June 2003.
- Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about the ones you need and ignore the rest.
- Using spring can significantly speed up development time

Spring modules

- Spring framework is divided into modules which makes it really easy to pick and choose in parts to use in any application:
  - **Core**: Provides core features like DI (Dependency Injection), Internationalisation, Validation, and AOP (Aspect Oriented Programming)
  - **Data Access**: Supports data access through JTA (Java Transaction API), JPA (Java Persistence API), and JDBC (Java Database Connectivity)
  - **Web**: Supports both Servlet API (Spring MVC) and of recently Reactive API (Spring WebFlux), and additionally supports WebSockets, STOMP, and WebClient
  - **Security**: provides support for securing your application and implementing authentication and authroziation
  - **Testing**: Wide support for unit and integration testing through Mock Objects, Test Fixtures, Context Management, and Caching
  - See the projects/modules spring providers: https://spring.io/projects


Spring Framework Runtime
Data Access/Integration
JDBC
ORM
OXM
JMS
Transactions
WEB
(MVC / Remoting)
Web
Servlet
Portlet
Struts
AOP
Aspects
Instrumentation
Spring Core Container
Core
Beans
Context
Expression Language
Test

Inversion of control (IoC)

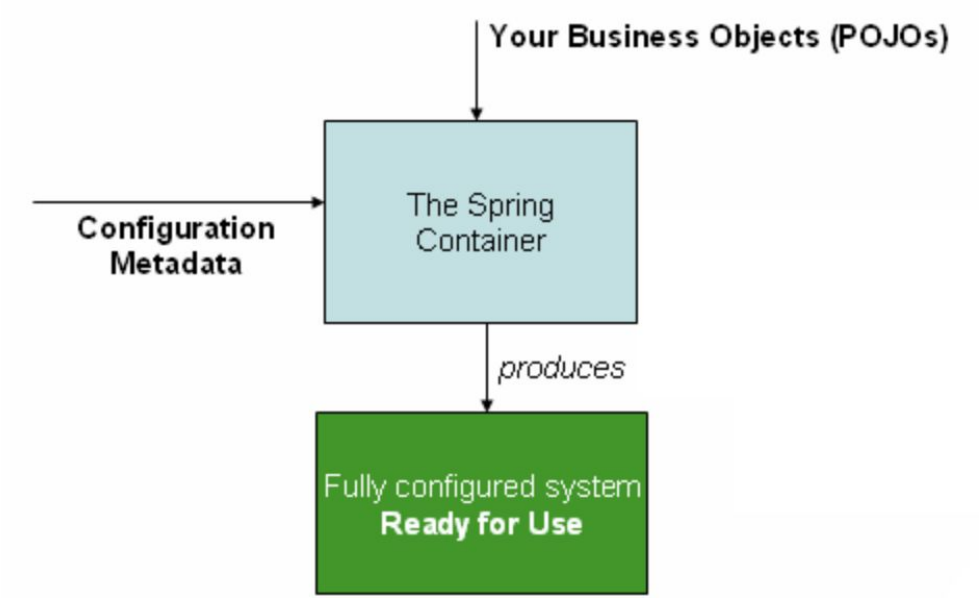


Figure 1. The Spring IoC container

Inversion of Control (IoC)

- Central in the Spring is its **Inversion of Control** container
- Based on “Inversion of Control Containers and the Dependency Injection pattern”
- Provides centralized, automated configuration, managing and wiring of application Java objects
- Container responsibilities:
  - creating objects,
  - configuring objects,
  - calling initialization methods
  - passing objects to registered callback objects
  - etc
- All together form the object lifecycle which is one of the most important features

Java objects that are managed by the Spring IoC container are referred to as **beans**

Defining beans in application context

- Component scan - spring scans scans all classes in our project and finds, initializes all beans
- Using annotations: **@Component**, **@Service**, **@Repository**, **@Controller**
- Java (in **@Configuration** classes using **@Bean** annotation)

Bean scopes

- Singleton
- Propotype
- https://howtodoinjava.com/spring-core/spring-bean-scopes/

Dependency injection - Autowiring/injecting beans

Beans can be autowired via:

- Constructor (preferred way)
- Setter
- Property

Spring boot

- Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".
- They take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.
- https://spring.io/projects/spring-boot

Spring boot features

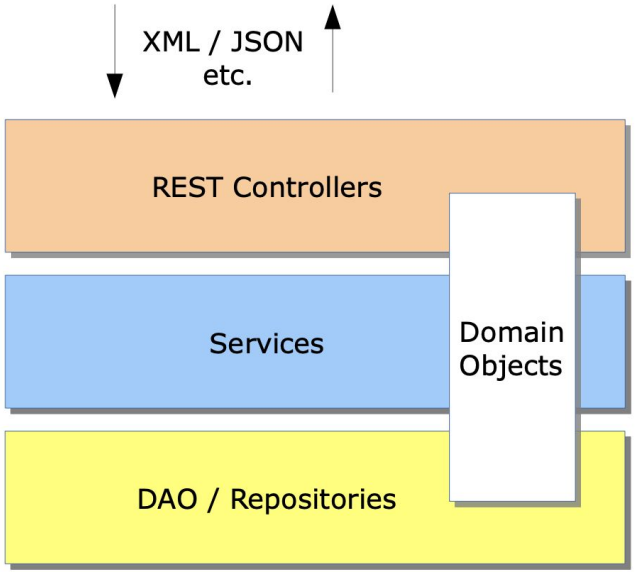
- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

Start your spring boot project

- See https://spring.io/quickstart

Typical web application

- Typical web application architecture
- REST Controllers provide CRUD interface to clients
- DAO provide CRUD interface to DB



More resources

- https://docs.spring.io/spring-framework/docs/current/reference/html/index.html
- https://spring.io/guides
- https://www.baeldung.com/spring-tutorial