

The background is a light blue gradient. In the top left corner, there are several white, realistic-looking water droplets of various sizes. In the bottom right corner, there are several light blue, realistic-looking water droplets of various sizes. The text is centered in the middle of the image.

SPRING

BY PANKAJ JAGASIA

WHAT IS A FRAMEWORK?

- a **software framework** is an abstraction in which software providing generic functionality can be selectively changed by user code, thus providing application specific software.
- A software framework is a universal, reusable software platform used to develop applications, products and solutions.
- Software frameworks include support programs, compilers, code libraries, an application programming interface (API) and tool sets that bring together all the different components to enable development of a project or solution.
- Contd.....

CONTD...

- Frameworks contain key distinguishing features that separate them from normal libraries:
- Inversion of control - in a framework, unlike in libraries or normal user applications, the overall program's flow of control is not dictated by the caller, but by the framework.^[1]
- Default behavior - A framework has a default behavior. This default behavior must actually be some useful behavior and not a series of no-ops.
- Extensibility - a framework can be extended by the user usually by selective overriding or specialized by user code providing specific functionality.
- Non-modifiable framework code - the framework code, in general, is not allowed to be modified. Users can extend the framework, but not modify its code.

GENERAL TYPES OF SOFTWARE FRAMEWORK

- [Application Framework](#), used to implement the standard structure of an application for a specific operating system.
- [Web Application Framework](#), for development of dynamic websites, web applications, and web services
- [Multimedia Framework](#), handles media on a computer and through a network

WHAT IS SPRING ?

- Invented by Rod Johnson in 2003 as OPEN SOURCE initially called Interface21
- Initially called as Interface 21 and then renamed to Spring Source and later to Spring
- The organisation responsible for Spring was Spring Source but was later taken over by Pivotal (VMWare owners @ \$420 m in 2009)
- In 2013 VMWare and its parent financial company EMC Corporation created joint venture Pivotal Software

contd...

WHAT IS SPRING ? contd...

- Spring came into being in 2003 as a response to the complexity of the early J2EE specifications
- Spring **integrates** with carefully selected individual specifications from the Java EE
 - Servlet API
 - WebSocket API
 - Concurrency Utilities
 - JSON Binding API
 - Bean Validation
 - JPA
 - JMS
 - JTA
- The Spring Framework also supports the Dependency Injection ([JSR 330](#)) and Common Annotations ([JSR 250](#)) specifications,

contd...

WHAT IS INVERSION OF CONTROL ?

- **Inversion of Control (IOC)** also known as **Dependency Injection (DI)** allows the application to be controlled by a container rather than the programmer defining the details of object creations
- **Container** a concealed environment that follows a set of specifications to define the life of various objects and the methods acting upon them
- Using the concept of IOC the programmer is only focused on the core business logic and the configuration needed to work with the objects and is kept unaware of the other aspects such as instantiation, releasing the objects/resources etc

contd...

WHAT IS INVERSION OF CONTROL ?

- DI can be defined as a process whereby objects define their dependencies, that is, the other objects they work with, only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method.
- The container then injects those dependencies when it creates the bean. This process is fundamentally the inverse, hence the name Inversion of Control (IoC), of the bean itself controlling the instantiation or location of its dependencies by using direct construction of classes, or a mechanism such as the Service Locator pattern.

KEY FEATURES OF SPRING ?

- **Inversion of Control (IOC) or Dependency Injection** is key strength of spring
- Mainly used for providing a loose coupling for various objects in an application without having to make any modifications in your application
- **Spring is a light weight Dependency Injection and Aspect Oriented Programming (AOP) Container and Framework**

CONTAINER ?

- **Container** *a concealed environment that follows a set of specifications to define the life of various objects and the methods acting upon them*
- The org.springframework.beans and org.springframework.context are the basis of the Spring IOC Container
- Two types of containers in spring
 1. *BeanFactory*
 2. *ApplicationContext*

contd...

BEAN FACTORY ?

- The BeanFactory interface provides an advanced configuration mechanism capable of managing any type of object.

APPLICATION CONTEXT ?

- The ApplicationContext is a sub-interface of the BeanFactory.
- It adds easier integration with
 - Spring's AOP features
 - message resource handling (for use in internationalization)
 - event publication
 - application-layer specific contexts such as the WebApplicationContext

SO WHAT DOES THE APPLICATION CONTEXT FINALLY DO ?

- The interface `ApplicationContext` represents the Spring IoC container and is responsible for :-
 - instantiating
 - Configuring
 - assembling the aforementioned beans.
- The container gets its instructions on what objects to instantiate, configure, and assemble by reading **configuration metadata**.
- The configuration metadata is represented in XML, Java annotations or Java code. It allows you to express the objects that compose your application and the rich interdependencies between such objects.

APPLICATION CONTEXT

1. ClasspathXmlApplicationContext (xml to be present in the classpath)
2. FileSystemXmlApplicationContext (refers to the system file path as mentioned)
3. XmlWebApplicationContext (web applications)
4. AnnotationConfigApplicationContext (config file free)

BEAN SCOPE

1. Singleton (only one object irrespective of the number of request)
2. Prototype a **new object** FOR each REQUEST
3. Request
4. Session
5. Global

** (3,4,5 USED FOR WEB APP)

AUTOWIRING

- Its a feature in Spring that will be used to automatically populate the properties of bean without we having to explicitly provide information
- Types of auto wiring
 1. ByName
 2. ByType
 3. Constructor

DISADVANTAGE OF AUTOWIRING

1. Once specified as auto-wiring cannot be defined as property in xml
2. If two properties are of the same type spring will raise an exception in bytype
3. Wiring is not clear as it takes place implicitly.,

WHATS NEW IN SPRING 4.X?

1. Removed deprecated packages and methods
2. Java 8 support (Lambda, JODA, 1.8 Byte Code format)
3. Recommends Java 6+ to be used
4. Java EE 6 and 7 support included hence now we can use Servlet API 3.0 + (though back compat for 2.5 is still supported)
5. Groovy Bean Definition DSL is now possible (external bean config file can now be defined in Groovy)

CONTD....

CONTD...

6. Core container improvements

1. Spring now treats [Generic Types as a form of qualifier](#) when injecting beans
2. Development of custom annotations that [expose specific attributes from the source annotation](#) is now possible.
3. Beans can now be ordered when they are [autowired into lists and arrays](#). Both the `@Ordered` annotation and `Ordered` interface are supported
4. The `@Lazy` annotation can now be used on injection points, as well as `@Bean` definitions.
5. The [@Description annotation has been added](#) for developers using java-based configuration
6. A generalized model for [conditionally filtering beans](#) has been added via the `@Conditional` annotation.
7. There is managed time zone support across the framework now, e.g. On `LocaleContext`.

CONTD....

CONTD...

7. General web improvements

1. Deployment to Servlet 2.5 servers remains an option, but Spring Framework 4.0 is now focused primarily on servlet 3.0+ environments
2. WebSockets is now supported
3. You can use the new **@RestController** annotation with spring MVC applications, removing the need to add **@ResponseBody** to each of your **@RequestMapping** methods.
4. The **AsyncRestTemplate** class has been added, allowing non-blocking asynchronous support when developing rest clients.

CONTD....

CONTD...

8. Websocket, sockjs, and STOMP messaging (Simple Text based Messaging Protocol)

A new spring-websocket module provides comprehensive support for websocket-based, two-way communication between client and server in web applications.

WHATS NEW IN SPRING 5 ?

1. JDK baseline update for spring framework 5.0

- The entire Spring Framework 5.0 codebase runs on Java 8. Therefore, Java 8 is the minimum requirement to work on Spring Framework 5.0.

2. Core Framework revision

- The Core Spring Framework 5.0 has been revised to utilize the new features introduced in Java 8
- Based on Java 8 reflection enhancements, method parameters in Spring Framework 5.0 can be efficiently accessed.
- Core Spring Interfaces now provide selective declarations built on Java 8 default methods.
- @Nullable and @NotNull annotations will explicitly mark nullable arguments and return values.

Contd...

CONTD...

3. Core container updates

- Spring Framework 5.0 now supports candidate component index as an alternative to **Classpath scanning**. (Reading entities from the **index** rather than scanning the classpath does not have significant differences for small projects with less than 200 classes)
- An application build task can define its own **META-INF/spring.Components** file for the current project. At compilation time, the source model is introspected and JPA entities and spring components are flagged.

Contd...

CONTD...

3. Core Container updates

- Implementation of functional programming style in `@GenericApplicationContext` and `@AnnotationConfigApplicationContext`
- Consistent detection of transaction, caching, and async annotations on interface methods.
- Xml configuration namespaces streamlined towards unversioned schemas.

CONTD...

CONTD...

4. Functional programming with kotlin

- Spring Framework 5.0 introduces support for JetBrains **KOTLIN** language. Kotlin is an object-oriented language supporting functional programming style. Kotlin runs on top of the JVM but is not limited to it.

5. Reactive Programming Model

- Spring releases the new Reactive Stack Web Framework
- Being fully reactive and non-blocking, this stack is suitable for event-loop style processing that can scale with a small number of threads.
- [Reactive streams](#) is an api specification developed by engineers from Netflix, Pivotal, TypeSafe, Red Hat, Oracle, Twitter, and Spray.IO

Contd...

CONTD...

- Spring Framework 5.0 has a new spring-webflux module that supports reactive HTTP and WebSocket clients. Spring Framework 5.0 also provides support for reactive web applications running on servers which include REST, HTML, and websocket-style interactions.
- There are two distinct programming models on the server-side in spring-webflux
 1. Annotation-based with @Controller and the other annotations of spring MVC
 2. Functional style routing and handling with Java 8 Lambda
contd...

CONTD...

6. Testing improvements

- Spring Framework 5.0 fully supports [JUnit 5 Jupiter](#) to write tests and extensions in JUnit 5. In addition to providing a programming and extension model, the Jupiter sub-project provides a test engine to run jupiter-based tests on Spring.

7. Library support

- a. [Jackson 2.6+](#)
- b. [Ehcache](#) 2.10+ / 3.0 ga
- c. [Hibernate 5.0+](#)
- d. [Jdbc 4.0+](#)
- e. [Xmlunit 2.X+](#)
- f. [Okhttp 3.X+](#)
- g. [Netty 4.1+](#)

DISCONTINUED IN SPRING 5.X

At the API level, spring framework 5.0 has discontinued support for the following packages

- beans.factory.access
- jdbc.support.nativejdbc
- mock.staticmock
- web.view.tile2m
- orm.hibernate3 and orm.hibernate4

Contd....

DISCONTINUED IN SPRING 5.X

Spring framework 5.0 has also discontinued support for the following libraries

- Portlet
- Velocity
- JasperReports
- XMLBeans
- JDO
- Guava