



Introduction to Spring

By Rahul Barve



Objectives

- What is Spring
- Why Spring
- Spring Features
- Spring Modules
- Spring Architecture
- First Example



What is Spring?



What is Spring?

- Spring is a lightweight dependency injection, aspect-oriented container and framework.



What is Spring?

- Spring is a Container as it creates the Java Components for your application and manages their life-cycle.



What is Spring Framework?

- It is lightweight in terms of size and overhead.
- It is distributed in a single ZIP file.



Why Spring Framework?



Why Spring Framework?

- Java components running inside Spring container have NO dependency on Spring specific Classes and Interfaces.



Why Spring Framework?

- Spring enables to build applications from “plain old Java objects” (POJOs) and to apply enterprise services to POJOs



Spring Features



Spring Features

- Loose Coupling
- Dependency Injection
- Aspect Oriented Programming
- Data Access Support
- Support for Enterprise Services
- Integration Support
- MVC Support



Spring Features

- Loose Coupling
 - Applications built using Spring are loosely coupled as it is not required to modify the Java source code in order to adopt changes in the configuration.



Spring Features

- Dependency Injection

- A technique through which, the dependent objects are given to the components so that components need not create dependent objects themselves.



Spring Features

- Dependency Injection

- ```
public class Employee {
 private Address addr;
 //Some Code
}
```



## Spring Features

- At the time of creating Employee, the container injects the inner object (Address) inside the main component (Employee).



# Spring Features

- Dependency Injection
  - Dependency Injection (DI) is also known as Inversion Of Control (IoC) as now the control is with the Container to create the Dependent Object and inject it inside the Component Object.





# Spring Features

- Aspect Oriented Programming
  - AOP is a programming model that promotes separation of business logic from the system concerns such as logging, transaction management, security, persistence etc.



# Spring Features

- Aspect Oriented Programming
  - The code that implements these system wide concerns is NOT duplicated across multiple components.
  - The component development is simple as they can focus now only upon core business logic.



# Spring Features

- Data Access Support
  - Spring abstracts away the common code like opening and closing connections, so that the database code can be clean and simple.



# Spring Features

- Data Access Support
  - Spring doesn't attempt to implement its own ORM solution, but provides hooks into several popular ORM frameworks like Hibernate, JPA, iBATIS etc.



# Spring Features

- Support for Enterprise Services
  - An application built using Spring can acquire various enterprise level services like:
    - Transaction Management
    - Persistence
    - Asynchronous Messaging
    - Security
    - Task Scheduling



# Spring Features

- Integration Support
  - Spring provides support for integration with other technologies like EJB, JNDI, Web Services etc.
  - Already implemented services can be consumed very efficiently using Spring.



# Spring Features

- MVC Support
  - Spring comes with its own MVC framework and even it can integrated with existing popular MVC frameworks like Struts, JSF, Tapestry etc.



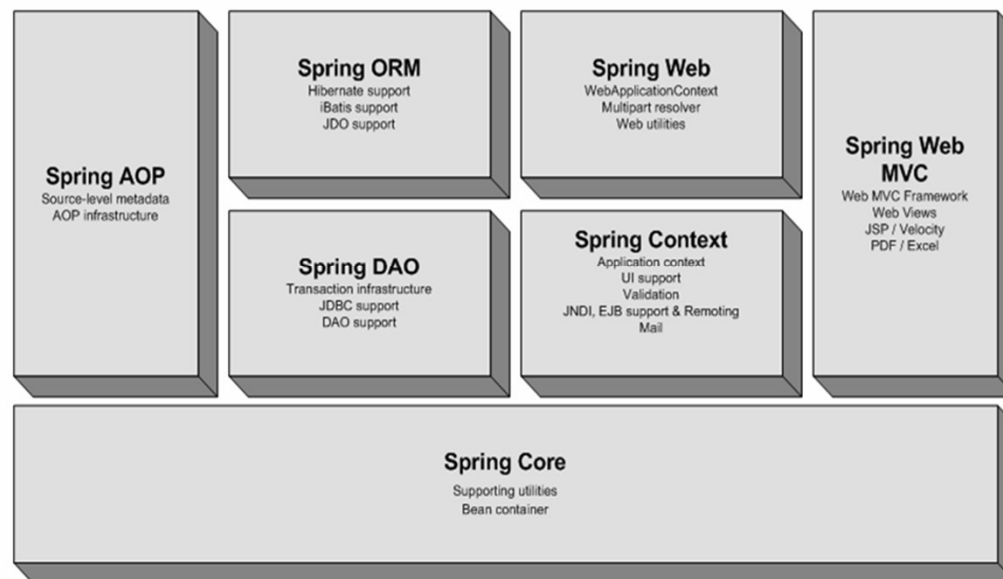
# ApplicationContext

- Built on the top of core container.
- Provides support for internationalization, application life cycle events and validation.



# Spring Framework – Core Modules

## Spring Architecture





# Let's Summarize

- What is Spring
- Why Spring
- Spring Features
- Spring Modules
- Spring Architecture
- First Example