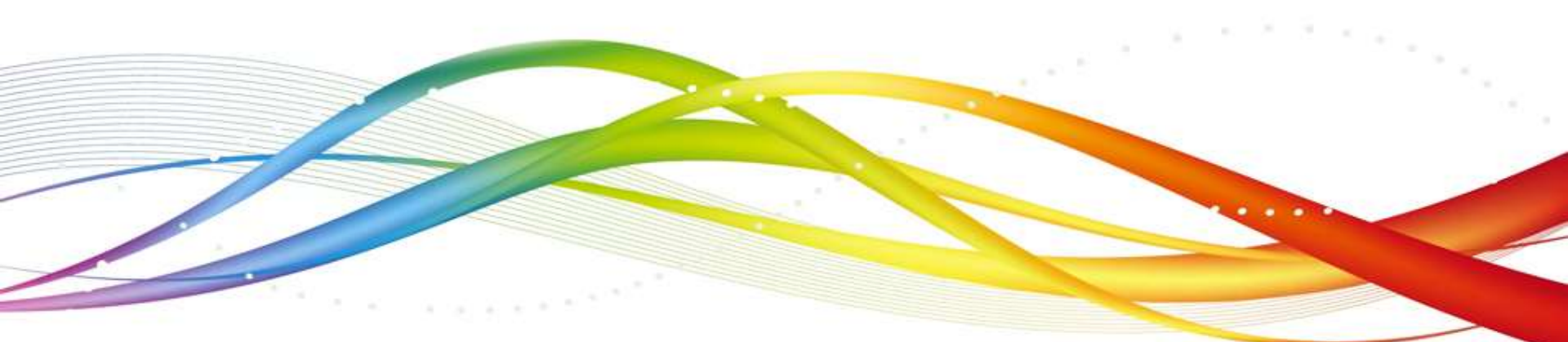




# Introduction to Spring



# Agenda

---

## Introduction to Spring

# Objectives

---

- At the end of this module, you will be able to understand:
  - Spring as Framework
  - Features of Spring
  - Architecture of Spring
  - Various Modules of Spring
  -

# What is Spring ?

---

- Spring 1.0 – released in March,2004 (by Rod Johnson & Juergen Holler), as a J2EE Application Framework.
- The current version is Spring Framework 4.0, which was released in December 2013
- Is a popular open source **application framework** that can make J2EE development easier by enabling a **POJO**-based programming model
- Consists of
  - a container
  - a framework for managing components
  - and a set of snap-in services for
    - web user interfaces
    - Transactions
    - and persistence

# Why Spring ?

---

- The essence of spring is in providing enterprise services to Plain Old Java Objects (POJO's)
- Applications built using Spring are very easy to unit test
- Spring can eliminate the need to use a variety of custom properties file formats, by handling configuration in a consistent way throughout applications and projects
- Spring can provide an alternative to EJB that's appropriate for many applications
- Spring provides a consistent framework for data access, whether using JDBC or an O/R mapping product such as TopLink, Hibernate or a JDO implementation.

# Spring == J2EE Application Server ?

---

- Spring is NOT a J2EE Application Server
- Spring can nicely integrate with J2EE Application Servers (or any Java Environment)
- Spring can elegantly replace the services traditionally provided by J2ee Application Server
- Spring provides elegant integration points with :
  - JDO
  - EJB
  - RMI
  - Web Services
  - JMS
  - Hibernate

# Features

---

- It is a lightweight framework
- There is no dependency on the framework
- Spring does not reinvent the wheel. Instead it makes all the existing solutions easier to use
- Spring is based on **Dependency Injection** flavor of **Inversion of Control**
- Spring includes a proxy **based AOP (Aspect Oriented Programming)** framework
- Spring does not provide its own ORM mapping framework.
  - Spring integrates well with all leading O/R mapping frameworks like hibernate, TopLink, JDP, Apache OJB, etc.
- It also integrates with a variety of web frameworks like struts, webwork, Spring MVC, Tapestry, JSP, etc.

# Features

- Spring is a major open source framework that aims to make J2EE development easier. Spring offers services throughout an application and not merely in a single architectural tier. The essence of spring is in providing enterprise services to Plain Old Java Objects (POJO's). It is a lightweight framework (does not have any high startup times, wont involve huge binary dependencies, will run in any environment). Spring provides an abstraction layer over JDBC that is simpler and less error prone to use than JDBC. It provides a transaction abstraction that can sit over JTA (Java Transaction API), global transactions, local transactions using JDBC, Hibernate, JDO or any other data access API's. Spring provides POJO based remoting over a variety of protocols including RMI, IIOP and other web services protocols. Spring provides support for sending and receiving JMS messages.
- The best part of using Spring is that the code need not be aware of Spring framework. Hence there is no dependency on the framework
- Spring is based on Dependency Injection flavour of Inversion of Control. This means that the framework code invokes application code, coordinating overall workflow rather than the application code invoking the framework code. Spring includes a proxy based AOP (Aspect Oriented Programming) framework.



# Architecture of Spring

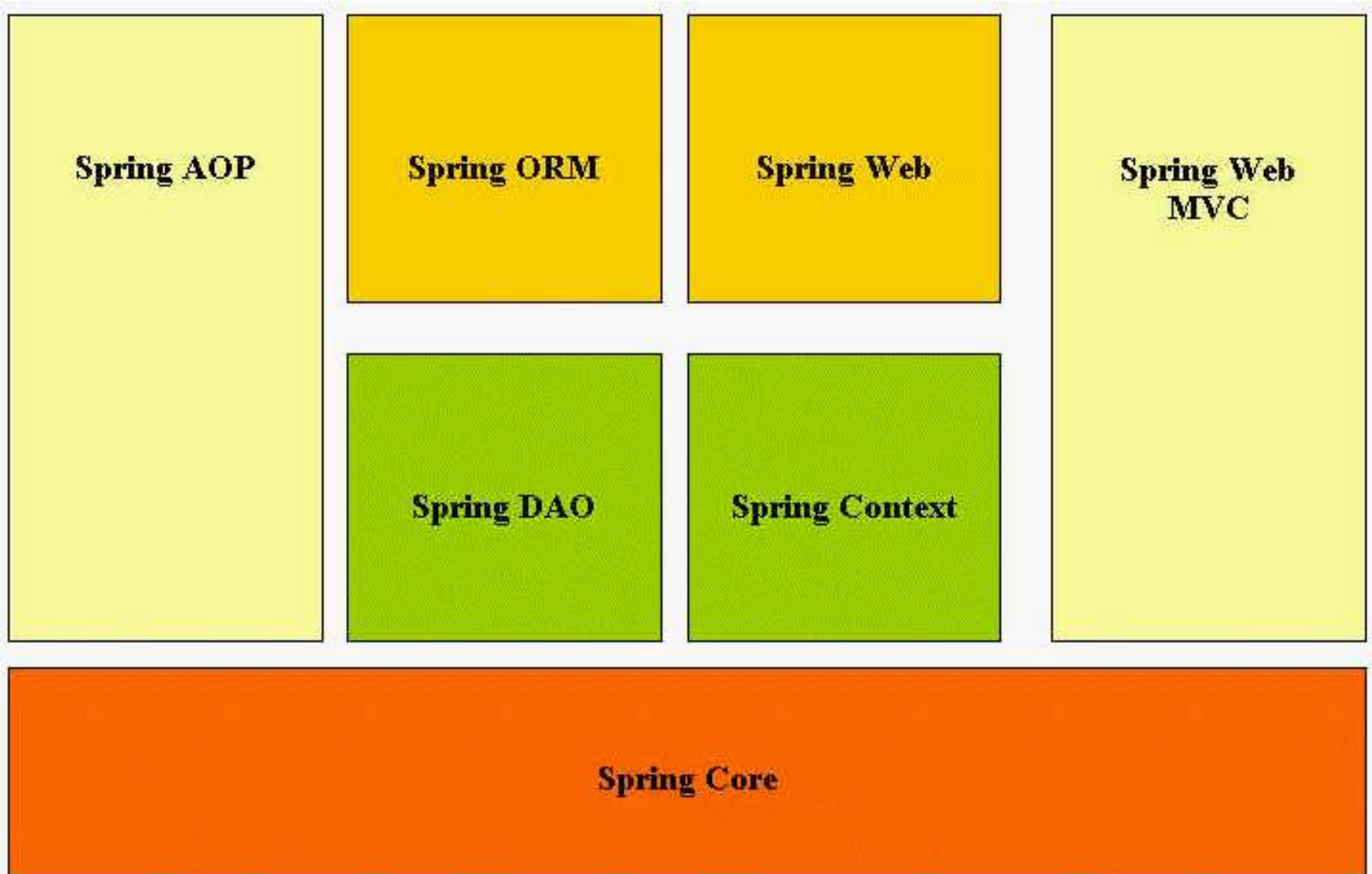
---

- Spring is well-organized architecture consisting of seven modules.
- Modules in the Spring framework are:
  - Spring AOP
  - Spring ORM
  - Spring Web
  - Spring DAO
  - Spring Context
  - Spring Web MVC
  - Spring Core

# Architecture of Spring

- AOP is used to provide declaration services to objects. It allows us to think about the concerns or aspects of the system like transaction management, logging, failure monitoring, etc.  
Spring does not provide its own ORM mapping framework. Spring integrates well with all leading O/R mapping frameworks like hibernate, TopLink, JDP, Apache OJB, etc. It also integrates with a variety of web frameworks like struts, webwork, Spring MVC, Tapestry, JSP, etc.
- Modules in the Spring framework are:
- **Spring AOP**  
One of the key components of Spring is the *AOP framework*. AOP is used in Spring:
  - To provide declarative enterprise services, especially as a replacement for EJB declarative services. The most important such service is *declarative transaction management*, which builds on Spring's transaction abstraction.
  - To allow users to implement custom aspects, complementing their use of OOP with AOP

# Architecture of Spring (Contd.).



# Architecture of Spring (Contd.).

---

- **Spring Jmx (spring-jmx.jar)**
  - **Spring Portlet (spring-portlet.jar)**
  - **Spring Remoting (spring-remoting.jar)**
  - **Spring Struts (spring-struts.jar)**
  - **Spring Support (spring-support.jar)**
  - **Spring Toplink (spring-toplink.jar)**
  - **Spring Web (spring-web.jar)**
  - **Spring Aspects (spring-aspects.jar)**
- 
- Classpath:
  - Application requires include any of these jar Files in its classpath, depending on the module(s) which are required. But, what will happen if an Application is in need of the functionality provided by all the various modules. Should it define entries for all the Jar Files in its class-path? Spring provides a smart solution for this need, as it comes with a Jar File called spring.jar which is a combination of all the modules.

# Spring Framework Components

---

## **Spring AOP**

To provide declarative enterprise services, especially as a replacement for EJB declarative services.

The most important such service is declarative transaction management, which builds on Spring's transaction abstraction.

To allow users to implement custom aspects, complementing their use of OOP with AOP

## **Spring ORM**

The ORM package is related to the database access. It provides integration layers for popular object-relational mapping APIs, including JDO, Hibernate and iBatis.

## **Spring Web**

The Spring Web module is part of Spring's web application development stack, which includes Spring MVC.

## **Spring DAO**

The DAO (Data Access Object) support in Spring is primarily for standardizing the data access work using the technologies like JDBC, Hibernate or JDO.

# Spring Jar files

---

- The name of the Spring module along with the jar file name (which is available in the SPRING\_HOME\dist\modules) is listed below.
  - **Spring Web MVC (spring-webmvc.jar)**
  - **Spring Aop (spring-aop.jar)**
  - **Spring Beans (spring-beans.jar)**
  - **Spring Context (spring-context.jar)**
  - **Spring Core (spring-core.jar)**
  - **Spring Dao (spring-dao.jar)**
  - **Spring Hibernate (spring-hibernate3.jar)**
  - **Spring Ibatis (spring-ibatis.jar)**
  - **Spring Jca (spring-jca.jar)**
  - **Spring Jdbc (spring-jdbc.jar)**
  - **Spring Jdo (spring-jdo.jar)**
  - **Spring Jms (spring-jms.jar)**
  - **Spring Jms (spring-jpa.jar)**

# Know your knowledge ...

---

- What is Spring?
  1. J2EE Design pattern
  2. J2EE Application Server
  3. J2EE Web Framework
  4. J2EE Application Framework
- Which module of Spring provides integration layer with Hibernate?
  1. Spring DAO
  2. Spring ORM
  3. Spring Context
  4. None
- Which module of Spring provides Dependency Injection feature?
  1. Spring Context
  2. Spring Core
  3. Spring DAO

# Summary

---

- In this module, we have learnt:
  - Spring as Framework
  - Features of Spring
  - Architecture of Spring
  - Various Modules of Spring





**Thank you**

