

Spring Framework



Agenda..

- Introduction to Spring
- Overview of Spring Architecture
- Spring IOC
- Basic Example (Spring Environment setup and Bean scopes)
- AOP Overview
- Spring MVC architecture
- Integration of Spring MVC with Hibernate
- Examples..

Introduction to Spring



- The **Spring Framework** is an open source application development **framework**
- Spring Provides comprehensive infrastructural support for developing enterprise Java applications very easily
- Spring is a lightweight framework. It can be thought of as a framework of frameworks because it provides support to various frameworks such as Struts, Hibernate, EJB, JSF etc.

Benefits of Using Spring Framework



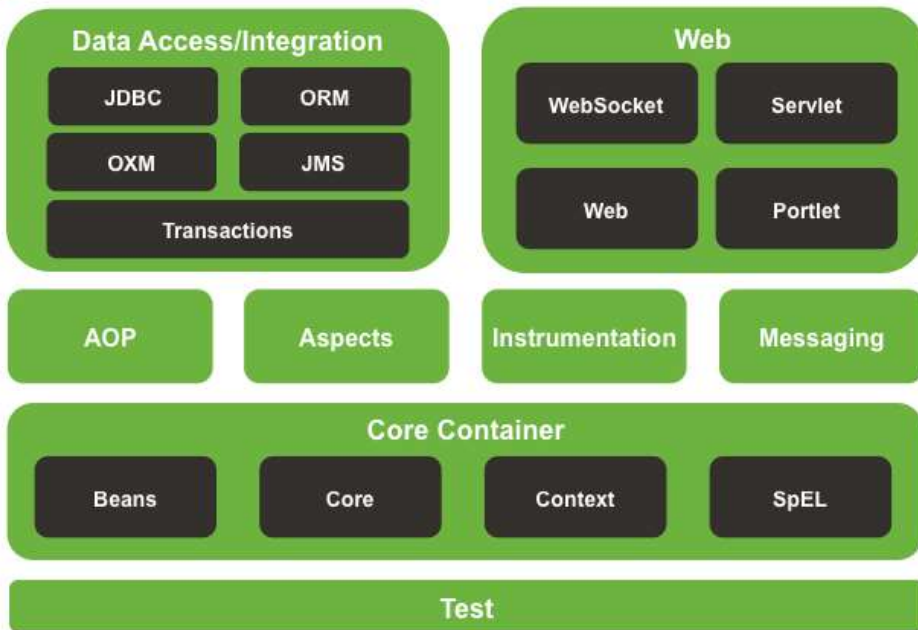
- Develop enterprise-class applications using POJOs
- Spring makes use of some of the existing technologies like ORM frameworks, Logging frameworks, JEE, JDK Timers etc..
- Good web MVC framework
- Nice technical api exception handling
- Best Transaction Management
- Light weight comparing EJBs
- Provides good Testing support
- Good security and logging service etc..



Overview of Spring Architecture



Spring Framework Runtime



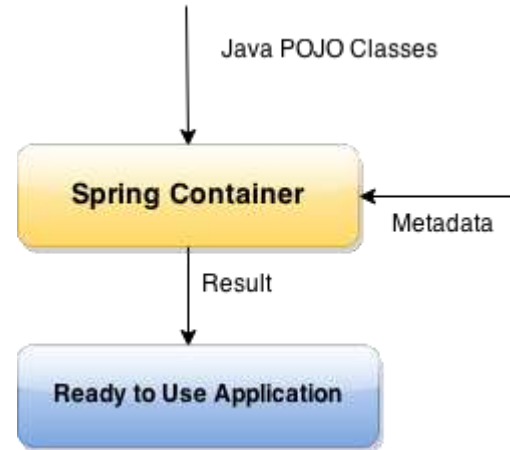
IOC Container



→ The core of the Spring Framework is its **Inversion of Control** (Ioc) container.

The container will create the objects, wire them together, configure them, and manage their complete lifecycle from creation till destruction.

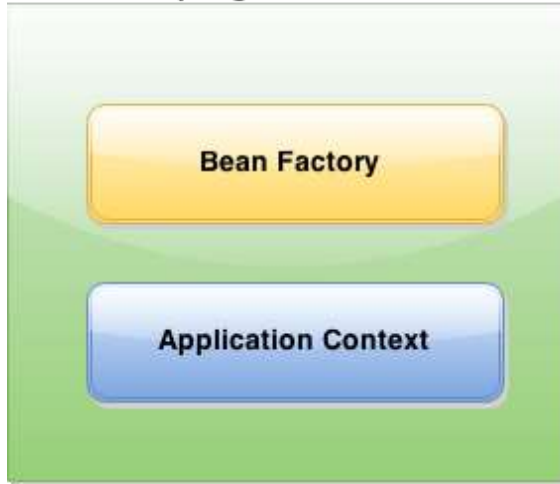
- ❖ To instantiate the class
- ❖ To configure the object
- ❖ To assemble the dependencies between the objects



continued..



Spring containers



1. `Resource resource=new ClassPathResource ("applicationContext.xml");`
2. `BeanFactory factory=new XmlBeanFactory(resource);`

```
ApplicationContext context = new  
ClassPathXmlApplicationContext("applicationContext.xml");
```

Dependency Injection



In Spring framework, Dependency Injection (DI) design pattern is used to define the object dependencies between each other.

Spring framework provides two ways to inject dependency

1. Setter Injection
2. Constructor Injection

Setter Injection

Constructor Injection

Bean Definition & Scopes



A bean is an object that is instantiated, assembled, and otherwise managed by a Spring IoC container.

Basic bean scopes

- **Singleton:** Only once for spring container
- **Prototype:** New bean created with every request or reference.

Web-aware context bean scopes

- **Request:** New bean for servlet request
- **Session:** New bean for session
- **Global session:** New bean for global HTTP session

Possible Bean Definition



A class with no dependencies

```
<bean id="service" class="example.ServiceImpl"/>
```

Results in (via Reflection):

```
ServiceImpl service = new ServiceImpl();
```

ApplicationContext

```
service -> instance of ServiceImpl
```

Spring - Java Based Configuration



```
@Configuration
public class HelloWorldConfig {
    @Bean
    public HelloWorld helloWorld(){
        return new HelloWorld();
    }
}
```

```
<beans>
  <bean id="helloWorld" class="com.
    tutorialspoint.HelloWorld" />
</beans>
```



```
ApplicationContext factory=new ClassPathXmlApplicationContext("applicationContext.xml");
HelloWorld helloWorld = ctx.getBean("helloWorld");
```



```
ApplicationContext ctx = new AnnotationConfigApplicationContext(HelloWorldConfig.class);
HelloWorld helloWorld = ctx.getBean(HelloWorld.class);
```

