



Spring Introduction



Presented by Hosein Zare
Twitter: @zare88r





- Spring فریم ورکی برای ساخت هر نوع اپلیکیشنی در جاوا میباشد شامل برنامه های standalone ، web ، JEE
- این فریم ورک lightweight (سبک وزن) میباشد نه از لحاظ حجم کلاس ها بلکه از نظر فلسفه این فریم ورک که به مینیمال بودن تأثیر آن بر کد و منطق اصلی برنامه، اهمیت میدهد (Spring Core)



تاریخچه

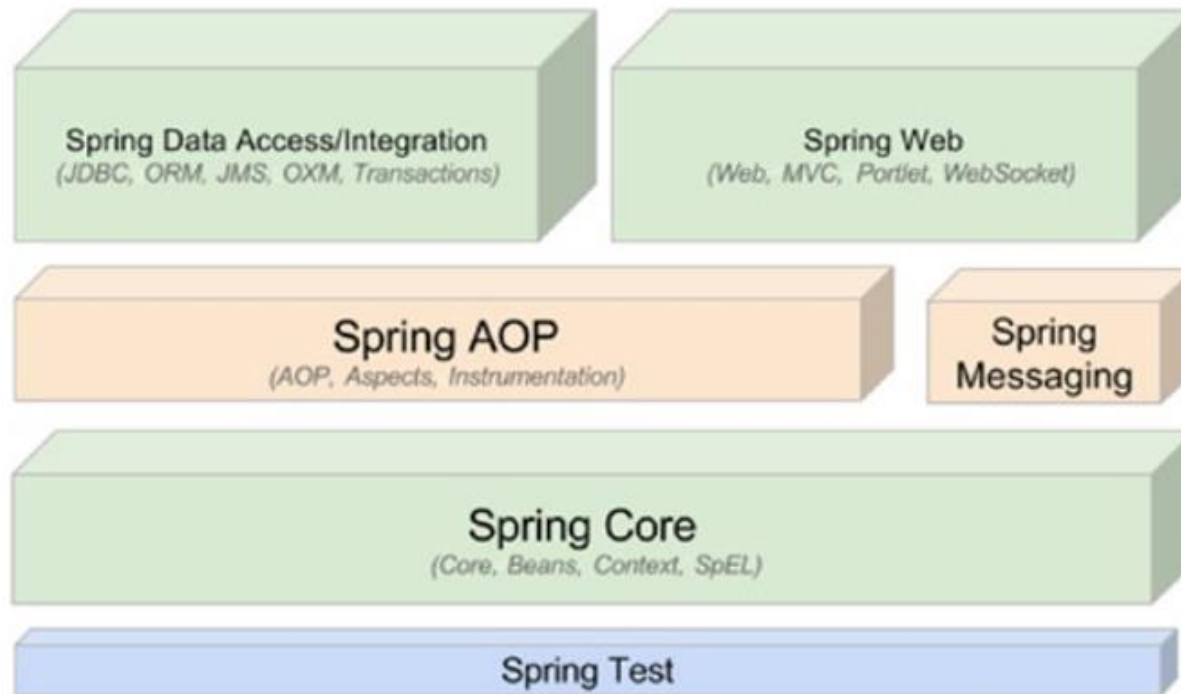
- اولین بار این فریم ورک در کتابی به نام

Expert One-on-One: J2EE Design and Development by Rod Johnson (Wrox, 2002)

معرفی شد.

- آخرین ورژن حال حاضر نسخه 5 میباشد که در تاریخ July 2016 عرضه شد

معماری





IOC به عنوان یک اصل طراحی

- Inversion of Control به این معناست که موجود دیگری اشیاء را برای شما بسازد. این موجود دیگر میتواند یک container باشد به طور مثال Java Servlet ها توسط برنامه نویس ساخته نمیشدند
- به علاوه این container وظیفه مدیریت چرخه حیات اشیاء نیز نیز می باشد
- به طور نمونه Servlet Container و Spring Beans



الگوی طراحی DI

- Dependency Injection الگوی طراحی میباشد که هدف آن جدا کردن رفتار از تحلیل نیازمندی هاست.

```
//Hardcoded dependency
public class MyClass {
    private MyDependency myDependency = new MyDependency();
}

//Injected dependency
public class MyClass {
    private MyDependency myDependency;
    public MyClass(MyDependency myDependency){
        this.myDependency = myDependency;
    }
}
```



IOC & DI

- با DI به تنهایی مجبوریم تمامی Dependency ها را در محلی دیگر تدارک ببینیم
- با ترکیب کردن IOC & DI به راحتی میتوانید bean های موجود در spring container را در bean های دیگر inject کرد

@Component

```
public class MyClass {
```

@Autowired

```
private MyDependency myDependency;
```

```
}
```



Aspect-Oriented Programming

- هدف از برنامه‌نویسی جنبه‌گرا، مستقل کردن وظایف (در قالب ماژول‌ها) است به‌طوری‌که کمترین تداخل را در یکدیگر داشته باشند تا نهایتاً بتوان از آن‌ها در برنامه‌های دیگر نیز استفاده نمود.
- وظیفهٔ اهراز هویت در یک برنامه را می‌توان به عنوان یک جنبه در نظر گرفت و ماژول جداگانه‌ای برای آن ساخت تا از این پس در تمام پروژه‌های مشابه بتوان از آن استفاده کرد

AspectJ –

Spring AOP –



SpEL

- Spring Expression Language یک زبان استاندارد که اجازه دستیابی به اشیاء و bean ها را در زمان اجرا میدهد
 - این زبان روشی متحد را بین تمامی ماژول های اسپرینگ به اشتراک میگذارد
- ```
Parser.parseExpression("Officers['president'].PlaceOfBirth.City")
 .getValue(societyContext,String.class);
```



# امكانات دیگر

- Validation in Spring(JSR-303 (Bean Validation))
- Accessing Data
  - JDBC , Hibernate , JDO , JPA , Mongo , Neo4j , ...
- Object/XML Mapping
  - XML Binding (JAXB) , Castor, XStream, JiBX, XMLBeans
- MVC
  - View Technologies : JSP , Velocity , Freemarker , Thymeleaf , JSF , Struts , ...



# امكانات دیگر

- WebSocket Support (JSR-356)
- Remoting Support
  - RMI, JAX-WS, JMS, Advanced Message Queuing Protocol (AMQP)
- Mail Support (JavaMail Api)
- Job Scheduling Support
- Dynamic Scripting Support



spring  
boot

- هر چه تعداد ماژول های مورد نیاز در یک برنامه اسپرینگ بیشتر باشد پیکربندی و مدیریت dependency های آن مشکل تر میشود
- Spring boot پروژه ای جهت تسهیل تعریف و پیکربندی پروژه های spring معرفی میکند که بتوان برنامه مورد نظر را در کمترین زمان اجرا گرفت

# جایگزین های Spring

- JBoss Seam Framework
- Google Guice
- PicoContainer
- JEE 8 Container

