# Spring框架概述

## 1.1轻量级：与EJB对比，依赖资源少

## 1.2分层：

Web层：Struts,SpringMVC

Service层：Spring

Dao层： hibernate,mybatis

## 1.3Spring核心

Spring的核心是控制反转IOC和AOP

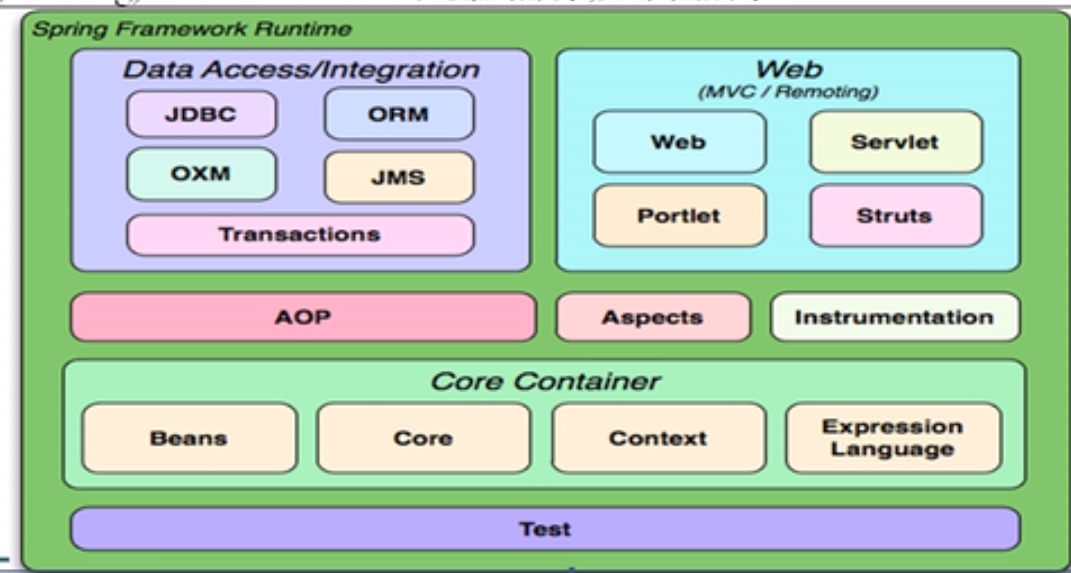
## 1.4Spring优点

a.方便解耦，易于开发

b.Spring就是一个工厂（容器），Spring工厂用于产生Bean

c.降低JavaEE开发中非常难用的一些API（JDBC、JavaMail、远程调用等），都提供了封装，使API难度降低

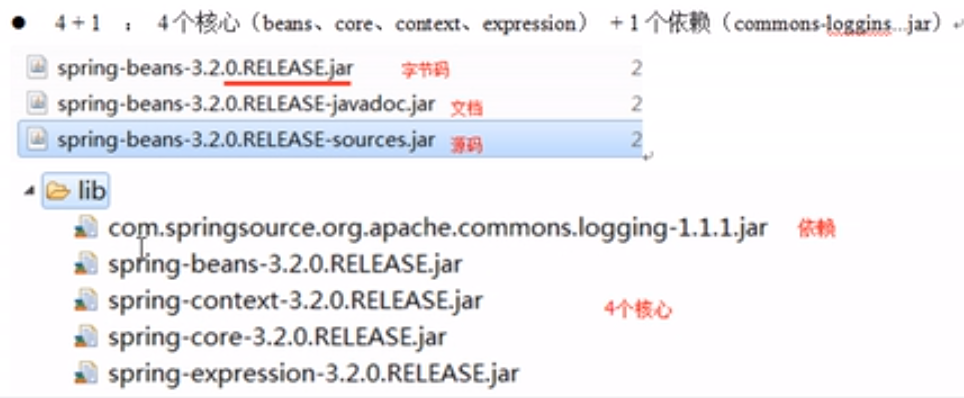
## 1.5Spring体系结构



核心容器：beans，core，context,sqel

# 入门案例：IOC

## 2.1导入jar包



## 2.2目标类

提供UserService接口和实现类

获得UserService实现类的实例

之前开发new一个对象即可。学习Spring之后，将由Spring创建对象实例—>Ioc控制反转，实现在XML文件中实现

## 2.3配置文件

位置：任意,开发中一般在classpath下(src)

名称：任意，开发中常用applicationContext.xml

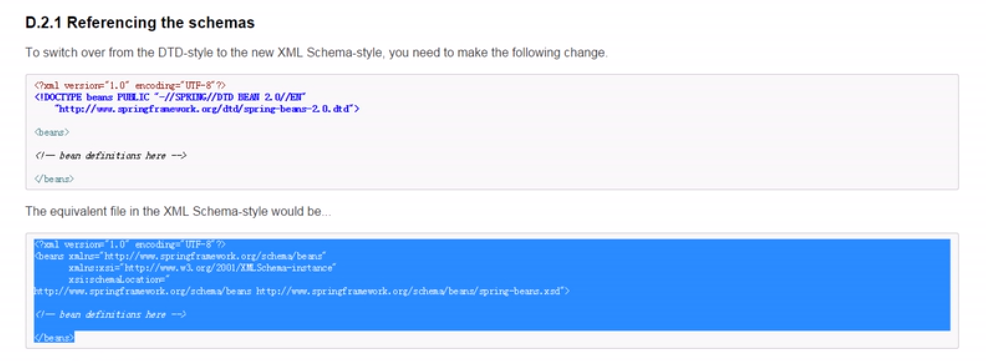
内容：添加scheme约束

约束文件位置：



**打开文件跳转网页，选择如图所示代码进行复制：**

**蓝色部分为scheme格式**



# 入门案例：DI

## DI Dependency Injection , 依赖注入

Is a : 是一个，继承

Has a :有一个，成员变量，依赖

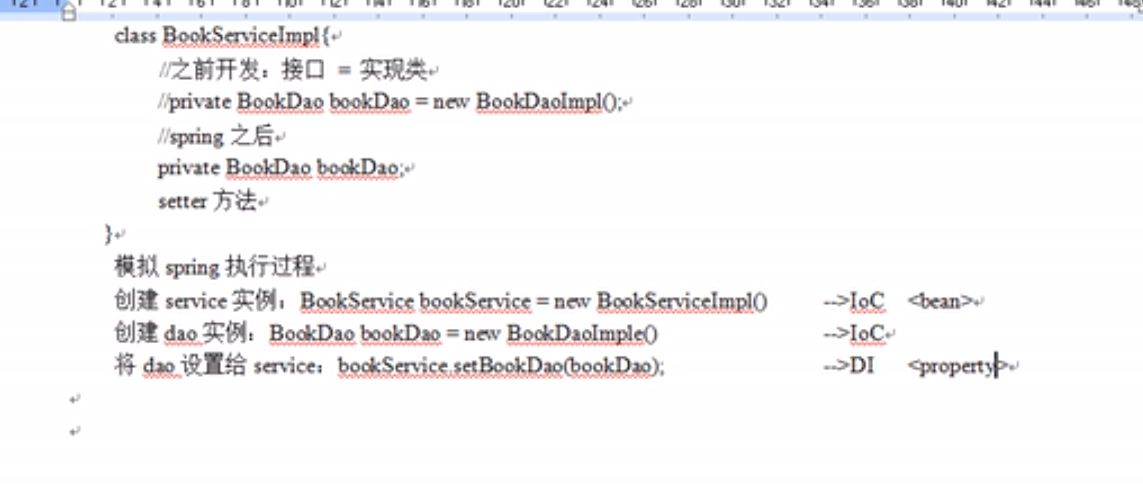
Class B{

Private A a ;//B类依赖A类

}

依赖：一个对象需要另一个对象实例设置

注入：通过setter方法进行另一个对象设置实例



## 3.1目标类

创建BookService

创建BookDao接口和实现类

将dao和service配置XML文件

使用API测试

### 3.1.1Dao

**public** **class** BookDaoIml **implements** BookDao {

@Override

**public** **void** addBook() {

System.***out***.println("di add BOOK!");

}

}

### 3.1.2Service

**public** **class** BookServiceImpl **implements** BookService {

//方式1：之前，接口=实现类

// private BookDao bookDao = new BookDaoIml();

//方式2：接口+setter

**private** BookDao bookDao;

**public** **void** setBookDao(BookDao bookDao) {

**this**.bookDao = bookDao;

}

@Override

**public** **void** addBook() {

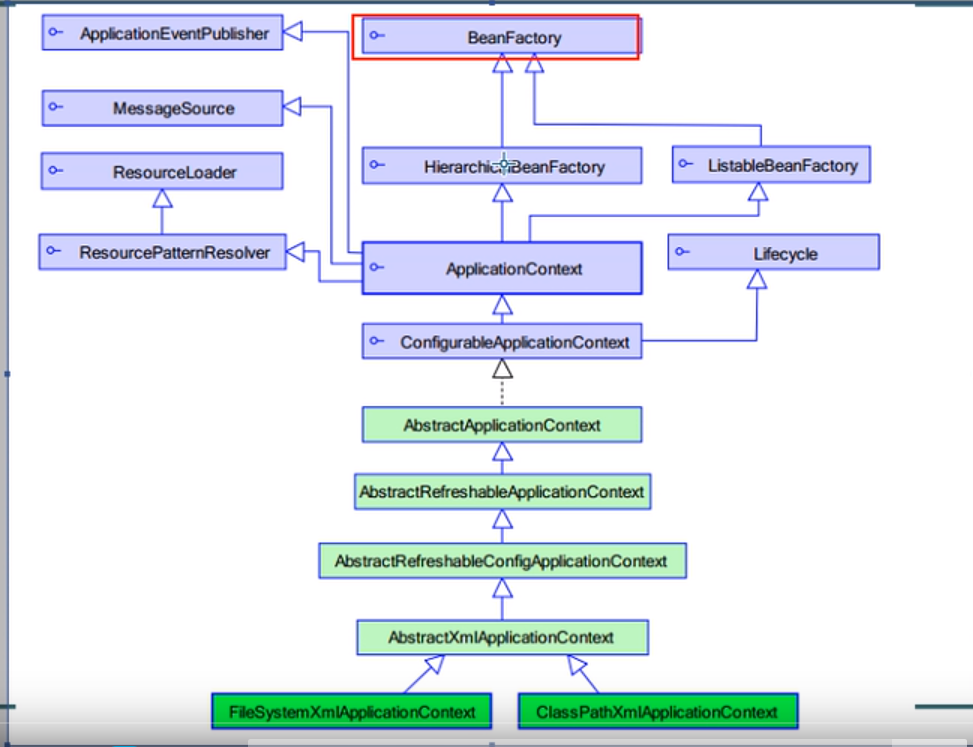
**this**.bookDao.addBook();

}

}

### 3.1.3 XML文件

# 核心API



Beanfactory:这是一个工厂，可以生产任意bean（配置文件被加载，对象就被new了）

特性：用getBean()才会加载Bean;

ApplicationContext： BeanFactory的子接口,功能更强大

ClassPathXmlApplicationContex 用于加载classpath(类路径，src)下制定的.xml文件

加载xml运行时位置🡪/WEB-INF/classes/..xml

FileSystemXmlApplicationContext 用于加载指定盘符下的xml

加载xml运行时为止🡪/WEB-IN/..xml

通过java.web.ServletContext.getRealPath() 获得盘符

## eclipse schema xml提示

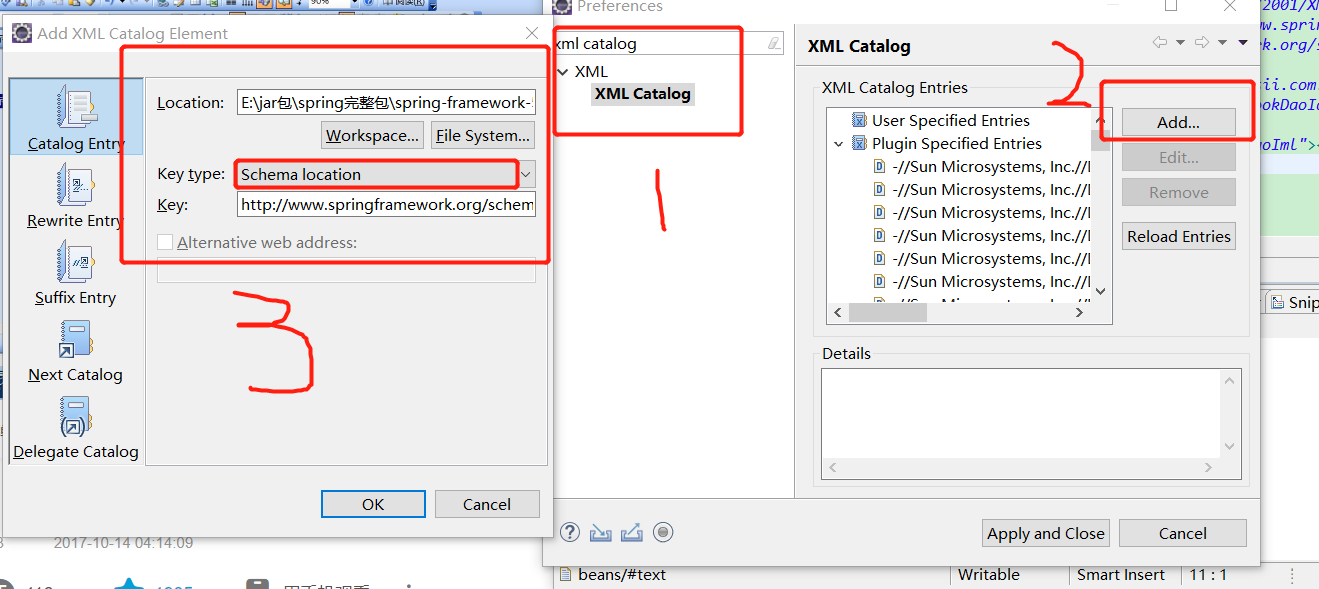
步骤一：确定xml位置

E:\jar包\spring完整包\spring-framework-5.0.0.RELEASE\schema\beans

*步骤二：复制路径*

[*http://www.springframework.org/schema/beans/spring-beans.xsd*](http://www.springframework.org/schema/beans/spring-beans.xsd)

步骤三：location放xml位置，key用步骤二的路径，keyType选择Scheme location



# 装配Bean基于xml文件

## 5.1实例化方法

### 5.1.1默认构造

### 5.1.2静态工厂

* 常用于Spring整合其他框架
* 静态工厂：用于生成实例化对象，所用的方法必须是静态方法

<bean id=” ” class=”工厂限定名” factory-method=”静态方法”>

## 5.2 作用域

## 5.3生命周期

# 装配Bean基于浅

# 整合junit

# 8.整合web