Spring框架

开源的轻量级框架

核心的两个部分

AOP，面向切面编程，扩展功能不是通过修改源代码来实现

IOC, 控制反转，把对象的创建交给spring管理

基于xml的配置文件

基于注解方式

DI 依赖注入，向类的属性设置属性值，

关系 DI不能单独使用，要依赖于IOC，在IOC的基础上进行操作

IOC底层原理

\*减弱类的创建与类的使用之间的耦合

主要技术：xml文件、dom4j解析xml文件、工厂模式、java反射机制

创建配置xml文件，配置要创建的类对象 <bean id = "user" class = "cn.itcast.ioc.User"></bean>

创建工厂类，使用dom4j解析配置文件，得到id对应的class属性值

利用反射机制创建对象

public static Item getItem(){

//dom4j解析配置文件，得到得到id对应的class属性值

String classValue = id对应的class属性值;

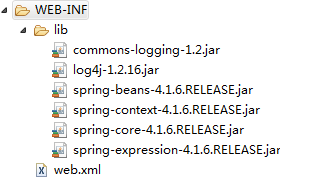
Class class = Class.forName(classValue);

Item item = class.newInstance();

return item;

}

导入jar包



引入schema约束

spring bean管理方式

spring bean管理就是通过配置文件创建对象

1、通过无参构造创建对象 (如果类没有无参构造会抛出异常)



2、通过静态工厂创建对象



3、通过实例工厂创建对象

Bean 标签的常用属性

Id 标签的名称

Class 类的全路径 qualified name

Name 标签的名称，但是可以包含特殊字符

Scope

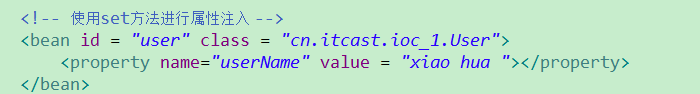
singleton 默认值，单例的

Prototype 多例的

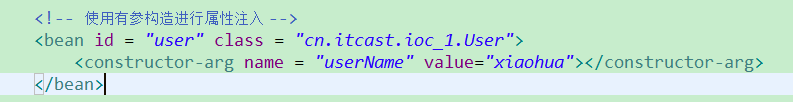


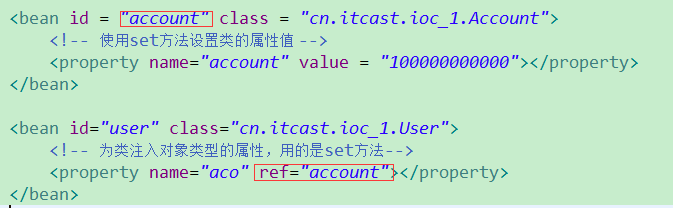
属性注入，设置类属性值

利用有setter方法为类注入属性



使用有参构造为类进行属性注入





Spring整合web项目

1、加载Spring核心配置文件

2、把对象的创建交给spring管理(new对象导致效率很低)

核心思想：将加载配置文件和创建对象的过程在服务器启动的时候完成，将压力交给服务器

ServletContext对象

监听器

\*服务器启动的时候，会为每个项目创建一个ServletContext对象，监听器会监听ServletContext对象何时创建的，并在创建ServletContext对象的时候，加载Spring核心配置文件，并根据配置文件创建对象，并将创建的对象放到ServletContext对象对象域里面(通过setAttribute方法)获取对象的时候到对象域里面获取(getAttribute)

框架吧这些事情都已经做好，需要做的是配置监听器，指定spring配置文件的位置

注解：有特殊标记的代码

注解方式: @注解名称(propertyName = propertyValue)

注解可以用在类、属性以及方法上面

1、在配置文件中配置注解扫描



2、注解创建对象



用在创建对象上的注解

@Component

@Controller

@Service

@Repository

目前这四个功能是一样的，为了以后功能扩展的方便

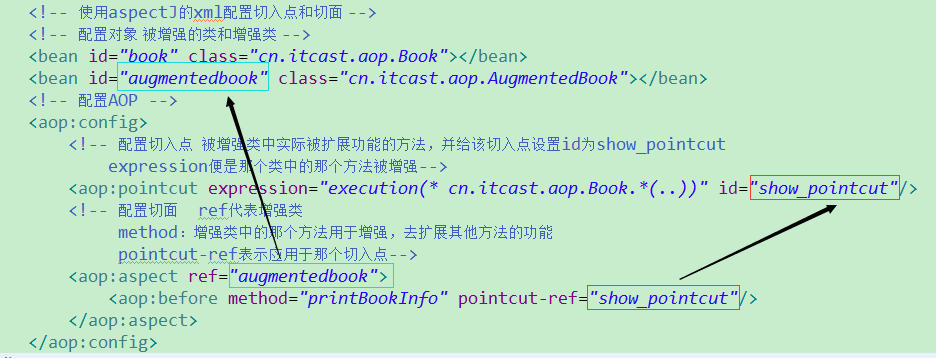
2、注解进行属性注入 使用注解方式进行属性注入的时候不需要set方法

为Server对象注入User对象的属性



Spring bean管理xml和注解方式的混合使用，一般创建对象用配置文件的方式，注入属性用注解的方式进行

AOP



Log4j 关于日志的工具

导入jar包，写配置文件 还可以设置日志的级别 info debug