IOC 控制反转=DI 依赖注入

* 反转获取对象的方式，从SpringIOC容器中获取，而不是自己创建new，然后setId赋值

一个bean就是一个对象

在applicationContext.xml中, <bean>创建对象，<property>给属性赋值

<bean id="ssss" class="org.linran.Student"> 包名.类名

<property name="id" value="12"></property>

<property name="name" value="zhangxin"></property>

</bean>

获取对象

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

Student s1=(Student)context.getBean("ssss");

给属性赋值方法1：setID方法

如果ssss的teacher属性是一个对象，则property要用ref而不是value

<property name="teacher" ref="teacher"></property>

<bean id="teacher" class="org.linran.Teacher">

<property name="name" value="liufurenlaoshi"></property>

</bean>

这种给属性赋值的方式，就是调用了setId, setTeacher方法。这方法名错一个字母都不行

Spring通过反射实现。通过类找方法

给属性赋值方法2：构造器

Student类中

public Student() {

super();

}

//这个没有参数的构造器得写，否则方法1中的就不能用了

public Student(int id, String name, Teacher teacher) {

super();

this.id = id;

this.name = name;

this.teacher = teacher;

}

applicationContext.xml中

<bean id="s2" class="org.linran.Student">

<constructor-arg value="11"></constructor-arg>

<constructor-arg value="寇鹏飞"></constructor-arg>

<constructor-arg ref="teacher"></constructor-arg>

</bean>

如果赋值和构造器参数顺序不一致，则

* 1.可以用index指定顺序

<constructor-arg ref="teacher" index="2"></constructor-arg>

<constructor-arg value="11" index="0"></constructor-arg>

<constructor-arg value="寇鹏飞" index="1"></constructor-arg>

* 2.可以用name指定参数名

<constructor-arg value="11" name="id"></constructor-arg>

<constructor-arg ref="teacher" name=""teacher"></constructor-arg>

<constructor-arg value="寇鹏飞" name="name"></constructor-arg>

* 3.用type指定参数

<constructor-arg value="11" type="int"></constructor-arg>

<constructor-arg ref="teacher" type="Teacher"></constructor-arg>

<constructor-arg value="寇鹏飞" type="String"></constructor-arg>

* 给属性赋值方法3：p命名空间
* Namespaces里勾选p – <http://www.springframework.org/schema/p>
* 创建对象时直接在标签里写p:id=, 如果是引用类型要加-ref

<bean id="s3" class="org.linran.Student" p:id="1" p:name="林然" p:teacher-ref="teacher">

</bean>

给list array set map注入，分别有各自的标签<list><set><map>

<property name="listProperty">

<list>

<value>string here</value>

</list>

</property>

<map>

<entry>

<key><value></value></key>

<value></value>

</entry>

</map>

<property>

<props>

<prop key="foot4">足球4</prop>

<props>

<property>

AOP

**搭建环境**

1.下载jar包

* 去官网下spring-framework-5.1.4.RELEASE

libs jar包 教程上是dics，应该是4以前的版本，现在变了

docs 文档

* aop.jar
* beans.jar
* context.jar
* core.jar
* expression.jar spring表达式
* 还有一个commons-logging.jar，需要另外下载，日志jar

新建java project——6个jar包复制到src文件夹——然后这6个右键，build path——add to build path——然后这6个jar包就变到了referenced libraries里边

* 编写配置文件

安装插件spring tool suite

springsource-tool-suite-3.9.7.RELEASE-e4.9.0-updatesite.zip

失败了，直接下载sts，相当于一个整合了spring的eclipse

新建java project——build path上一步——新建Spring bean configuration file, 命名为applicationContext

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">

</beans>

# 使用Spring开发Web项目

* SpringIOC容器初始化：

1.将容器中所有bean实例化为对象

2.将各个bean依赖的属性值注入

* Web项目初始化SpringIOC容器：

当服务Tomcat启动时，通过监听器将SpringIOC容器初始化一次

spring-web.jar提供

1.新建Dynamic Web Project（如果不勾选Web.xml，会在Web-Content/WEB-INF位置下少一个Web.xml文件，内容为下）

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>lll</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

</web-app>

2.把7个（6+1个spring-web.jar）jar包放进Web-Content/WEB-INF/lib目录下，add to build path

3.在web.xml里添加

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener.class</listener-class>

</listener>

这段代码配置spring-web.jar提供的监听器，此监听器可以在服务器启动时初始化SpringIOC容器

4.初始化容器还需要告诉容器（applicationContext）的位置

在ContextLoaderListener.class的父类ContextLoader.class的源码中有一行

**public** **static** **final** String ***CONFIG\_LOCATION\_PARAM*** = "contextConfigLocation";

CONFIG\_LOCATION\_PARAM就是一个属性，属性值为contextConfigLocation，保存着容器配置文件applicationContext.xml的位置

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:applicationContext.xml</param-value>

</context-param>

5.在Java Resources/src位置下new一个Spring Bean Configuration File。此时src就是classpath类路径

6.教程上还configure build path🡪add library🡪server runtime🡪Tomcat，没做这一步也行好像

拆分Spring配置文件

<param-value>

classpath: applicationContext.xml,

classpath: applicationContext-DAO.xml,

classpath: applicationContext-Service.xml

</param-value>

或者applicationContext-\*.xml

方法3

<import resource="applicationContext-\*.xml" />