

# 【java开发系列】—— spring简单入门示例

## 回顾

- 1 [JDK安装](#)
- 2 [Struts2简单入门示例](#)

### 前言

作为入门级的记录帖，没有过多的技术含量，简单的搭建配置框架而已。这次讲到spring，这个应该是SSH中的重量级框架，它主要包含两个内容：控制反转\依赖注入，和AOP面向切面编程。

## 1 控制反转IOC\依赖注入DI，因为翻译的不同，因此有两个名字。

**控制反转**意思就是说，当我们调用一个方法或者类时，不再有我们主动去创建这个类的对象，控制权交给别人（spring）。

**依赖注入**意思就是说，spring主动创建被调用类的对象，然后把这个对象注入到我们自己的类中，使得我们可以使用它。

举个简单的例子，程序猿加班了一个月，很累，想要放松下，于是去找人吃“麻辣烫”。

不使用spring的传统做法是，我们自己通过陌陌微信等神器，主动寻找目标，花费大量人力物力，达成协议后，申请“场所”办正事。

而使用spring的做法就很方便了，我们直接去某个场所，那个地方直接就有目标候着！

这是多么神奇的功能啊！



（此图涉及暴力，18岁以下自行负责）

## 2 AOP面向切面编程

我们把一个方法看做是一个切面，在这个切面的前后或者周围，都可以设置其他的处理方法，进行一些特殊的处理。

比如一个“吃蛋糕”的方法，在这个方法前需要“拆包装”，方法后需要“打扫卫生”，都可以通过这种编程方式来实现。

### 那么下面看下它的编码过程！

- 1 首先我们需要一个bean，用某种方法（设置注入需要有setter函数，构造注入需要有构造函数，相应的bean.xml配置文件也会不同）设定注入方式。
- 2 bean.xml，设置bean与类的关系，并关联默认的注入值。
- 3 获取bean.xml文件，创建实例对象，直接调用方法。

可以看到，我们开发的程序只有第三步的“创建实例对象，直接调用方法”，并没有为他进行初始化等工作，就可以直接调用它的方法，获取它的值。其实spring在程序初始化的时候，就会为我们把bean对应的对象注入进来，帮助我们完成初始化，因此我们只要通过一个引用对象，就可以直接调用了。

### 公告

昵称：xingoo

园龄：5年1个月

粉丝：2708

关注：66

+加关注

<	2017年10月			
日	一	二	三	四
24	25	26	27	28
1	2	3	4	5
8	9	10	11	12
15	16	17	18	19
22	23	24	25	26
29	30	31	1	2

### 最新随笔

1. 《湖畔》—— 读后总结
2. 程序猿的日常——SpringMVC与流程回顾
3. 为博客园添加github跳转按钮
4. TDDL与Spring Boot集成案例——跟踪与解决
5. NLP1 —— Python自然语言处理搭建
6. 个性化搜索的几个阶段
7. 自然语言处理扫盲·第四天——机器问答系统原理
8. 自然语言处理扫盲·第三天——情感分析原理
9. 自然语言处理扫盲·第二天——机器翻译原理
10. 自然语言处理扫盲·第一天——自然语言处理的背景、应用、推荐资源

### 随笔分类(780)

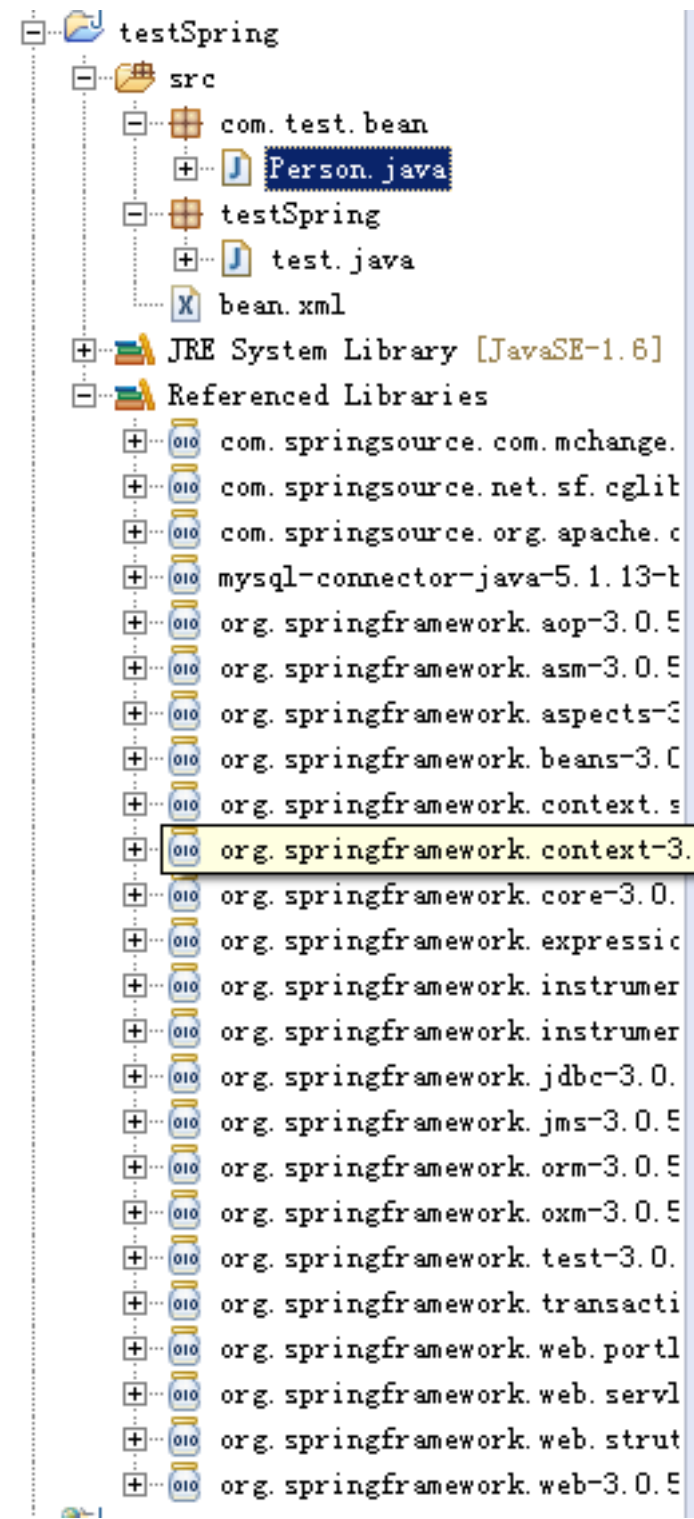
AngularJS(26)

Elasticsearch(53)

最后我们写一个简单的依赖注入的小样例！

1 需要的jar包和文件结构

jar包下载地址：http://pan.baidu.com/s/1sjDbzrR



2 bean.xml配置文件，放在src目录下

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns="http://www.springframework.org/schema/beans"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
  <bean id="person" class="com.test.bean.Person">
    <property name="name" value="xingoo"/>
    <property name="age" value="12"/>
  </bean>
</beans>
```

3 Person类，这里使用的是set设置注入方式

```
package com.test.bean;

public class Person {

    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String name) {
```

Hadoop(17)

Hbase(2)

Java(137)

JavaScript(71)

Kafka(2)

linux(29)

Logstash(16)

MongoDB(2)

Oozie(12)

Oracle(44)

Python(2)

Redis(1)

Ruby(8)

Scala(8)

Spark(19)

Spring(25)

Sqoop(3)

TensorFlow(1)

程序人生(118)

缓存系统(6)

机器学习(10)

全栈折腾(11)

软件考试(1)

设计模式(23)

数据仓库(9)

数学理论

```
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void info(){
        System.out.println("一起来吃麻辣烫！");
        System.out.println("name:"+getName()+" age:"+getAge());
    }
}
```



#### 4 test类



```
package testSpring;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.test.bean.Person;

public class test {
    public static void main(String[] args){
        ApplicationContext ctx = new ClassPathXmlApplicationContext("bean.xml");//读取bean.xml中的内容
        Person p = ctx.getBean("person",Person.class);//创建bean的引用对象
        p.info();
    }
}
```



#### 5 执行结果

Problems Javadoc Declaration Console

<terminated> test [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2014年9月10日 上午10:43:36)

2014-9-10 10:43:38 org.springframework.context.support.AbstractApplicationContext pr  
信息: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1  
2014-9-10 10:43:39 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loa  
信息: Loading XML bean definitions from class path resource [bean.xml]  
2014-9-10 10:43:40 org.springframework.beans.factory.support.DefaultListableBeanFact  
信息: Pre-instantiating singletons in org.springframework.beans.factory.support.Defau  
一起来吃麻辣烫!  
name:xingoo age:12

分类： Java

好文要顶 关注我 收藏该文

weibo qq

xingoo

关注 - 66

粉丝 - 2704

+加关注

17 0

推荐 反对

« 上一篇： 【java开发系列】—— struts2简单入门示例

» 下一篇： 【java开发系列】—— 集合使用方法



#1楼 2016-03-13 08:08 护身符 

按照你的代码敲得不知道为什么出错了，是配置错了吗？

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/commons/logging/LogFactory
at org.springframework.context.support.AbstractApplicationContext.<init>(AbstractApplicationContext.java:154)
at org.springframework.context.support.AbstractApplicationContext.<init>(AbstractApplicationContext.java:215)
at org.springframework.context.support.AbstractRefreshableApplicationContext.<init>(AbstractRefreshableApplicatio
nContext.java:88)
at org.springframework.context.support.AbstractRefreshableConfigApplicationContext.<init>(AbstractRefreshableCo
nfigApplicationContext.java:58)
at org.springframework.context.support.AbstractXmlApplicationContext.<init>(AbstractXmlApplicationContext.java:
61)
at org.springframework.context.support.ClassPathXmlApplicationContext.<init>(ClassPathXmlApplicationContext.jav
a:136)
at org.springframework.context.support.ClassPathXmlApplicationContext.<init>(ClassPathXmlApplicationContext.jav
a:83)
at testSpring.test.main(test.java:11)
Caused by: java.lang.ClassNotFoundException: org.apache.commons.logging.LogFactory
at java.net.URLClassLoader$1.run(Unknown Source)
at java.net.URLClassLoader$1.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
... 8 more
```

支持(0) 反对(0)

#2楼[楼主] 2016-03-13 09:51 xingoo 

@ 护身符

org/apache/commons/logging/LogFactory 你缺少commons-logging这个包。如果是自己的代码，可以把Logger去掉。如果是公司的代码，可以找找日志的相关资料，把缺少的jar包加进去。

支持(1) 反对(0)

#3楼 2016-04-17 18:02 T-mac\_wzn 

楼主，我感觉您觉得这个小例子不足以说明spring这个框架的优势；


您的这个例子，相比于直接使用构造方法，然后通过构造方法来调用，有什么优势呢？

```
public person （String name,int age） {
this.name=name;
this.age=age;
}

person p = new pperson("xingoo",12)
p.info();
```

本人初学者，希望楼主赐教；

支持(0) 反对(0)

#4楼[楼主] 2016-05-09 07:59 xingoo 

@ T-mac\_wzn

再创建单个对象的时候，看不出效果。但是在对象很多，甚至实现的方式很多的时候就有效果了。

基于你的例子，如果我需要去修改Person的属性和方法，是不是意味着也需要修改new那里的代码！

如果使用spring就不会了，bean类的实现---xml配置---业务实现，这三者之间是独立的，每次修改只需要修改其中的两者而已。

相当于把业务与真正的代码实现解耦了。

支持(1) 反对(0)

#5楼 2016-10-17 22:11 LAOK23 

指出楼主的几个问题吧：1、要是新手不引入lz网盘的包，而引用通常的spring框架的包，会提示commons-logging包缺少。2、要是新手全部复制bean文件的话，会报错，建议这句<?xml version="1.0" encoding="UTF-8"?>不要复制，不然有空格会报错。3、bean文件里面的id="person"应该为id="Person"。说得不对的地方敬请谅解。

支持(0) 反对(2)

#6楼 2016-11-16 11:29 举个栗子 

怎么测试的？  
作为WEB程序运行还是Application client Project？  
IDE新手求

支持(0) 反对(0)

#7楼[楼主] 2016-11-16 11:32 xingoo 

@ 举个栗子  
直接运行的main函数,application这种方式

支持(0) 反对(0)

#8楼 2016-12-26 01:57 我只个穷小子罢了 

@ LAOK23  
既然解耦合，xml中的id，就不能与类有任何关联，这个id是随便可以取得，您可以试试。id只要保证规范和唯一性就可以了

支持(0) 反对(0)

#9楼 2017-01-06 11:26 前尘往事 

啥

支持(0) 反对(0)

#10楼 2017-01-10 15:51 肆芳海木 

@ LAOK23  
person可以不改的吧？ test里面调用的时候一样就行了

支持(0) 反对(0)

#11楼 2017-03-17 21:56 努力的小蜗牛 

ApplicationContext ctx =new ClassPathXmlApplicationContext("bean.xml");

支持(0) 反对(0)

#12楼 2017-05-03 19:54 霖醇谷 

我也是按照你的代码，打的，这个错误怎么解决  
java.lang.NoSuchMethodError: main  
Exception in thread "main"

支持(0) 反对(0)

#13楼[楼主] 2017-05-03 20:26 xingoo 

@ 霖醇谷  
没有main方法.....

支持(0) 反对(0)

#14楼 2017-05-03 20:33 霖醇谷 

@ xingoo  
public class test {  
public static void main(String[] args) {  
ApplicationContext ctx = new FileSystemXmlApplicationContext(  
"applicationContext.xml");  
HelloWorld hello = (HelloWorld) ctx.getBean("HelloWorld");

```
hello.execute();
}
}
这不是main方法吗？
我写上了呀
```

支持(0) 反对(0)

#15楼 2017-09-07 11:34 1179922534 

@ 霖醇谷  
你web.xml里面可能没有配<context-param>的一些配置 然后看一下 你的xml文件是否 在资源包里面。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。