



链滴

# Spring\_IOC 个人理解整理

作者: [goker](#)

原文链接: <https://ld246.com/article/1606724726823>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# spring简单介绍之IOC

## Spring IOC

### 一. IOC (控制反转)

原来由java类, new()对象的方式;

把创建对象、调用对象的过程交付给spring, 由Bean来创建对象。

#### 底层原理

xml解析(dom4j)+工厂模式+反射

#### ApplicationContext(BeaFactory)接口

IOC容器的核心接口 (实例化、定位、配置应用程序中的对象, 建立对象的依赖)。

以下接口:

#### BeaFactory (顶级接口)

内置的使用接口, 只有简单的功能。

加载配置文件时, 不会创建对象, 在获取或使用对象时, 才创建对象。

#### ApplicationContext, 继承实现了BeaFactory

提供更强大的功能、更多的扩展。

在启动时, 就把配置文件中的Bean全部实例化。

### 二. DI (依赖注入)

有参构造: <constructor-org>

set方法: <property name="" value="">, 使用外部bean; 加上 ref=""

p:命名空间: <bean p:name="">

### 三. Bean

通过bean标签来创建对象

<bean id="" class="">

id唯一表示, class指定的类 (包名.类名)

#### 3.1. Bean 生命周期

1. 无参构造创建实例
2. set注入属性 <property>
3. 初始化 init()
4. 普通方法
5. 销毁 destroy()

## 3.2. Bean 作用域

### 1. 单例（默认） `singleton`

两个对象的地址是一样的，`<bean id="" class="" scope="singleton">`。

### 2. 原形，每次都会创建一个新的 `prototype`

`<bean id="" class="" scope="prototype">`

## 3.3. Bean 懒加载

默认加载配置文件时，就初始化所有的bean。

配置按需加载Bean：

`<beans default-init-lazy="true">`，在bean根目录的最后。

## 3.4. Bean 创建对象

### 1. 有参和无参

### 2. 静态工厂方法

### 3. 实例工厂方法

### 4. 实现 `FactoryBean`接口：简单的工厂模式+装饰模式，对象是有`getObject()`方法决定。

## 3.5. 集合/Utils

### 1. `utils`

```
<utils:list id="myList">
  <value>list1 </value>
  <value>list2 </value>
  <value>list3 </value>
</utils:list>
```

### 2. `List`和`Set`

```
<property name="list" ref="myList"/>
```

```
<property name="list" >
  <list>(set)
    <value>list1 </value>
  </list>
</property>
```

### 3. `Map`

```
<property name="map" >
  <map>
    <entry key="k1" value="v1"/>
  </map>
</property>
```

## 4. Property

```
<property name="prop">
  <props>
    <prop key="k1">v1 </prop>
  </props>
</property>
```

## 3.6. 外部配置文件（数据库连接）

```
<context:property-placeholder location="classpath:db.properties"/>

<bean id="" class="">
  <property name="dirver" value="{jdbc.dirver}">
    .....
</bean>
```

## 四. 自动装配

spring会自动根据匹配的属性进行注入。

- **byName**
- **byType**
- **no**
- **default**
- **constructor**

### 注解@

1. 使用在类、方法、属性上面
2. 类（4个语义化注解）

- **@Controller**; servlet
- **@Service**; 业务
- **@Repository**; Dao、Mapper
- **@Component**; 其他公共类

### 3. @AutoWire

Service层要使用Dao层的方法：

在service层中使用自动装配：

```
@AutoWire
private UserMapper userMapper;
```

### 注意

在.xml文件中要配置包扫描

```
<context:component-scan base-package="扫描的包名"/>
```