

# 模拟 Spring 的 bean 加载过程

作者: [woshiszz](#)

原文链接: <https://ld246.com/article/1570212248604>

来源网站: [链滴](#)

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



一天不学习就感觉浑身痒痒，忽然想写一篇东西来记录自己学习的点点滴滴，哈哈，学习之路还很长不要轻易说放弃。

下面通过工厂模式简单的对spring的bean加载过程进行模拟：

## 改造前：

### dao层

```
public class UserDaoImpl implements IUserDao {
    public void saveUser() {
        System.out.println("保存成功");
    }
}
```

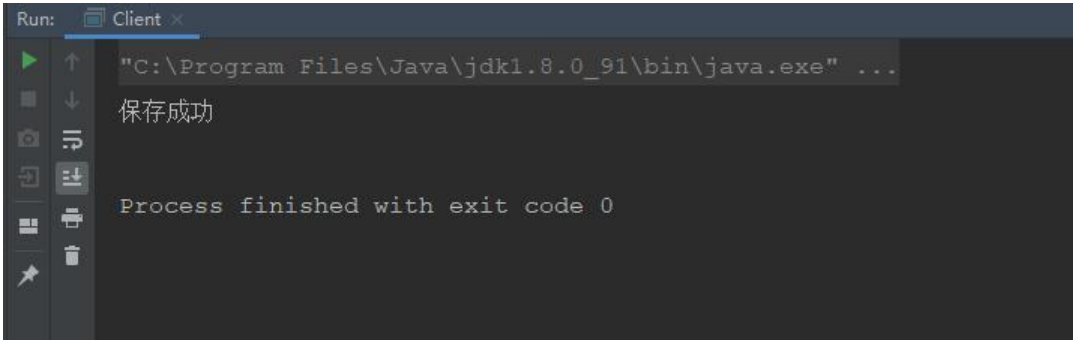
### service层

```
public class UserServiceImpl implements IUserService {
    private UserDaoImpl userDao=new UserDaoImpl();
    public void saveUser() {
        userDao.saveUser();
    }
}
```

### client

```
public class Client {
    public static void main(String[] args) {
        UserServiceImpl service = new UserServiceImpl();
    }
}
```

```
    service.saveUser();
}
}
```



## 改造后:

### bean.xml

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

    <bean id="userDao" class="com.gary.dao.impl.UserDaoImpl"> </bean>
    <bean id="userService" class="com.gary.service.impl.UserServiceImpl"> </bean>

</beans>
```

### BeanFactory

```
public class BeanFactory {
    /**
     * 定义一个map容器，存入要创建的对象名(key)与实例对象(value)
     */
    private static Map<String, Object> beans;
    private static SAXReader reader;
    static {
        try {
            beans = new HashMap<String, Object>();
            reader = new SAXReader();
            //从xml获取数据
            Document document = reader.read(BeanFactory.class.getClassLoader().getResourceAsStream("bean.xml"));
            //获取根元素,如<beans>
            Element root = document.getRootElement();
            //获取子元素,如<bean>
            List<Element> elements = root.elements();
            String key = null;
            String beanPath = null;
            Object value = null;
            for (Element element : elements) {
                //拿到xml id的值
                key = element.attributeValue("id");
```

```

        //拿到xml class的值
        beanPath = element.attributeValue("class");
        //通过反射实例化对象
        value = Class.forName(beanPath).newInstance();
        //把对象存入容器
        beans.put(key, value);
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

/**
 * 创建对象
 * @param beanName
 * @return
 */
public static Object getBean(String beanName) {
    return beans.get(beanName);
}
}

```

## dao层

```

public class UserDaoImpl implements IUserDao {
    public void saveUser() {
        System.out.println("保存成功");
    }
}

```

## service层

```

public class UserServiceImpl implements IUserService {
    private IUserDao userDao = (IUserDao) BeanFactory.getBean("userDao");
    public void saveUser() {
        userDao.saveUser();
    }
}

```

## client

```

public class Client {
    public static void main(String[] args) {
        IUserService service = (IUserService) BeanFactory.getBean("userService");
        service.saveUser();
    }
}

```

```
"C:\Program Files\Java\jdk1.8.0_91\bin\java.exe" ...  
保存成功  
Process finished with exit code 0
```