

简单的说，Spring就是通过**工厂+反射**将我们的bean放到它的容器中的，当我们想用某个bean的时候，只需要调用getBean("beanID")方法。

原理简单介绍：

Spring容器的原理，其实就是通过解析xml文件，或取到用户配置的bean，然后通过反射将这些bean挨个放到集合中，然后对外提供一个getBean()方法，以便我们获得这些bean。下面是一段简单的模拟代码：

```
package com.tgb.spring.factory;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.jdom.Document;
import org.jdom.Element;
import org.jdom.input.SAXBuilder;
import org.jdom.xpath.XPath;

public class ClassPathXmlApplicationContext implements BeanFactory {

    //容器的核心，用来存放注入的Bean
    private Map<String, Object> container = new HashMap<String, Object>
();

    //解析xml文件，通过反射将配置的bean放到container中
    public ClassPathXmlApplicationContext(String fileName) throws
Exception{
        SAXBuilder sb = new SAXBuilder();
        Document doc =
sb.build(this.getClass().getClassLoader().getResourceAsStream(fileName));

        Element root = doc.getRootElement();
        List list = XPath.selectNodes(root, "/beans/bean");

        //扫描配置文件中的bean
        for (int i = 0; i < list.size(); i++) {
            Element bean = (Element) list.get(i);
            String id = bean.getAttributeValue("id");
            String clazz = bean.getAttributeValue("class");
```

```

        Object o = Class.forName(clazz).newInstance();
        container.put(id, o);
    }
}

```

```

@Override
public Object getBean(String id) {
    return container.get(id);
}

```

```

}

```

首先声明一个存放bean的Map，然后通过jdom解析配置文件，循环遍历所有的节点，并通过反射将它们放到我们之前声明的Map中。然后提供一个getBean()的方法，让我们可以通过bean的Id来找到我们想要的bean。

下面是一个简单的xml配置文件：

```

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

    <bean id="E" class="com.tgb.spring.factory.England" />

    <bean id="S" class="com.tgb.spring.factory.Spain" />

    <bean id="P" class="com.tgb.spring.factory.Portugal" />

</beans>

```

客户端通过调用前面的ClassPathXmlApplicationContext，来加载上面的配置文件，然后就可以通过Id来获得我们需要的bean了：

```

package com.tgb.spring.factory;

public class Test {

    public static void main(String[] args) throws Exception {

        //加载配置文件
        BeanFactory f = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        //英格兰
        Object oe = f.getBean("E");
    }
}

```

```
Team e = (Team) oe;
```

```
e.say();
```

```
//西班牙
```

```
Object os = f.getBean("S");
```

```
Team s = (Team) os;
```

```
s.say();
```

```
//葡萄牙
```

```
Object op = f.getBean("P");
```

```
Team p = (Team) op;
```

```
p.say();
```

```
}
```

```
}
```

输出结果：

**England** ：我们是欧洲的中国队，不在乎这次小组没出线...

**Spain** ：我们是两届欧洲杯冠军、一届世界杯冠军！

**Portugal** ：我们的C罗一个顶十个！

其他代码：

```
//工厂接口
```

```
package com.tgb.spring.factory;
```

```
public interface BeanFactory {
```

```
    Object getBean(String id);
```

```
}
```

```
//Team接口
```

```
package com.tgb.spring.factory;
```

```
public interface Team {
```

```
    void say();
```

```
}
```

```
//英格兰
```

```
package com.tgb.spring.factory;
```

```
public class England implements Team{
```

```

    public void say() {
        System.out.println("England：我们是欧洲的中国队，不在乎这次小组没出线...");
    }
}

```

```
//西班牙
```

```
package com.tgb.spring.factory;
```

```
public class Spain implements Team{
```

```
    @Override
```

```
    public void say() {
```

```
        System.out.println("Spain：我们是两届欧洲杯冠军、一届世界杯冠军！");
```

```
    }
```

```
}
```

```
//葡萄牙
```

```
package com.tgb.spring.factory;
```

```
public class Portugal implements Team {
```

```
    @Override
```

```
    public void say() {
```

```
        System.out.println("Portugal：我们的c罗一个顶十个！");
```

```
    }
```

```
}
```

以上内容是对Spring的一个简单模拟，当然Spring远比这个要复杂的多，也强大的多，而且获取bean的方式也不止通过工厂这一种。这里只是做一个粗略的Demo说说自己对容器的简单理解，向Spring致敬。例子简陋，表达粗糙，欢迎拍砖交流。