

# Spring BeanFactory 容器

## Spring 的 BeanFactory 容器

这是一个最简单的容器，它主要的功能是为依赖注入（DI）提供支持，这个容器接口在 `org.springframework.beans.factory.BeanFactory` 中被定义。BeanFactory 和相关的接口，比如 BeanFactoryAware、DisposableBean、InitializingBean，仍旧保留在 Spring 中，主要目的是向后兼容已经存在的和那些 Spring 整合在一起的第三方框架。

在 Spring 中，有大量对 BeanFactory 接口的实现。其中，最常被使用的是 **XmlBeanFactory** 类。这个容器从一个 XML 文件中读取配置元数据，由这些元数据来生成一个被配置化的系统或者应用。

在资源宝贵的移动设备或者基于 applet 的应用当中，BeanFactory 会被优先选择。否则，一般使用的是 ApplicationContext，除非你有更好的理由选择 BeanFactory。

### 例子：

假设我们已经[安装 Eclipse IDE](#)，按照下面的步骤，我们可以创建一个 Spring 应用程序。

步骤	描述
1	创建一个名为 <code>SpringExample</code> 的工程并在 <code>src</code> 文件夹下新建一个名为 <code>com</code> 。
2	点击右键，选择 <code>Add External JARs</code> 选项，导入 Spring 的库文件，正如我们在 <code>Example</code> 章节中提到的导入方式。
3	在 <code>com.tutorialspoint</code> 文件夹下创建 <code>HelloWorld.java</code> 和 <code>MainApp.java</code>
4	在 <code>src</code> 文件夹下创建 Bean 的配置文件 <code>Beans.xml</code>
5	最后的步骤是创建所有 Java 文件和 Bean 的配置文件的內容，按照如下所示步骤

下面是文件 `HelloWorld.java` 的内容：

```
package com.tutorialspoint;
public class HelloWorld {
```

```
private String message;
public void setMessage(String message){
    this.message = message;
}
public void getMessage(){
    System.out.println("Your Message : " + message);
}
}
```

下面是文件 **MainApp.java** 的内容：

```
package com.tutorialspoint;
import org.springframework.beans.factory.InitializingBean;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.ClassPathResource;
public class MainApp {
    public static void main(String[] args) {
        XmlBeanFactory factory = new XmlBeanFactory
            (new ClassPathResource("Beans.xml"));
        HelloWorld obj = (HelloWorld) factory.getBean("helloWorld");
        obj.getMessage();
    }
}
```

在主程序当中，我们需要注意以下两点：

- 第一步利用框架提供的 **XmlBeanFactory()** API 去生成工厂 bean 以及利用 **ClassPathResource()** API 去加载在路径 CLASSPATH 下可用的 bean 配置文件。**XmlBeanFactory()** API 负责创建并初始化所有的对象，即在配置文件中提到的 bean。
- 第二步利用第一步生成的 bean 工厂对象的 **getBean()** 方法得到所需要的 bean。这个方法通过配置文件中的 bean ID 来返回一个真正的对象，该对象最后可以用于实际的对象。一旦得到这个对象，你就可以利用这个对象来调用任何方法。

下面是配置文件 **Beans.xml** 中的内容：

```
<?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-3.0.xsd">
```

```
<bean id="helloWorld" class="com.tutorialspoint.HelloWorld">  
  <property name="message" value="Hello World!"/>  
</bean>  
  
</beans>
```

如果你已经完成上面的内容，接下来，让我们运行这个应用程序。如果程序没有错误，你将从

控制台看到以下信息：

```
Your Message : Hello World!
```