

## 实验二.txt

### 实验二：真正的SpringIoC功能 (BeanFactory&ApplicationContext)

```
SpringDemo
|
|---src
|   |
|   |---tarena
|       |
|       |-----spring-config.xml
```

加入Spring2的相关的JAR包

步骤一，创建HelloBean类

```
public class HelloBean {
    private String words;

    public String getWords() {
        return words;
    }

    public void setWords(String words) {
        this.words = words;
    }
}
```

-----

步骤二，创建Spring配置文件

```
<?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-2.0.x
sd">
```

## 实验二.txt

```
<bean id="HelloBean" class="tarena.HelloBean">
    <property name="words">
        <value>达内，你好!</value>
    </property>
</bean>

</beans>
```

步骤三，创建测试类

```
public class Test {
```

```
    public static void main(String[] args) {

        Resource resource =
            new
FileSystemResource("src/tarena/spring-config.xml");

        BeanFactory beanFactory =
            new XmlBeanFactory(resource);

        HelloBean helloBean =

(HelloBean)beanFactory.getBean("HelloBean");

        System.out.println(helloBean.getWords());
    }
}
```

```
public class Test {

    public static void main(String[] args) {

        //一次性加载一个spring配置文件
        //ApplicationContext context =
        //    new
FileSystemXmlApplicationContext("src/tarena/spring-config.xml");

        //一次性加载多个spring配置文件
```

实验二.txt

```
//ApplicationContext context =
//      new
FileSystemXmlApplicationContext(
//      new
String[]{"src/tarena/spring-config.xml"}
//      );

//使用通配符*加载文件
ApplicationContext context =
      new
FileSystemXmlApplicationContext("src/tarena/spring*.xml");

HelloBean helloBean =

(HelloBean)context.getBean("HelloBean");

      System.out.println(helloBean.getWords());
}
}
```

-----

步骤四，结果  
达内，你好!