

【Spring】IntelliJ IDEA 搭建 Spring 环境

/本来在草稿箱写好了，忘记发就被冲掉了，重新再写一遍。

Spring 初探

文章目录

Spring 初探

在 IntelliJ IDEA 中创建 Spring 项目

一个简单的例子介绍框架的作用

那么什么时候 new 的对象呢

对 Spring 的作用的初步了解

开发环境：

IntelliJ IDEA 2016.2.5

JRE: 1.8.0_91-b14 amd64

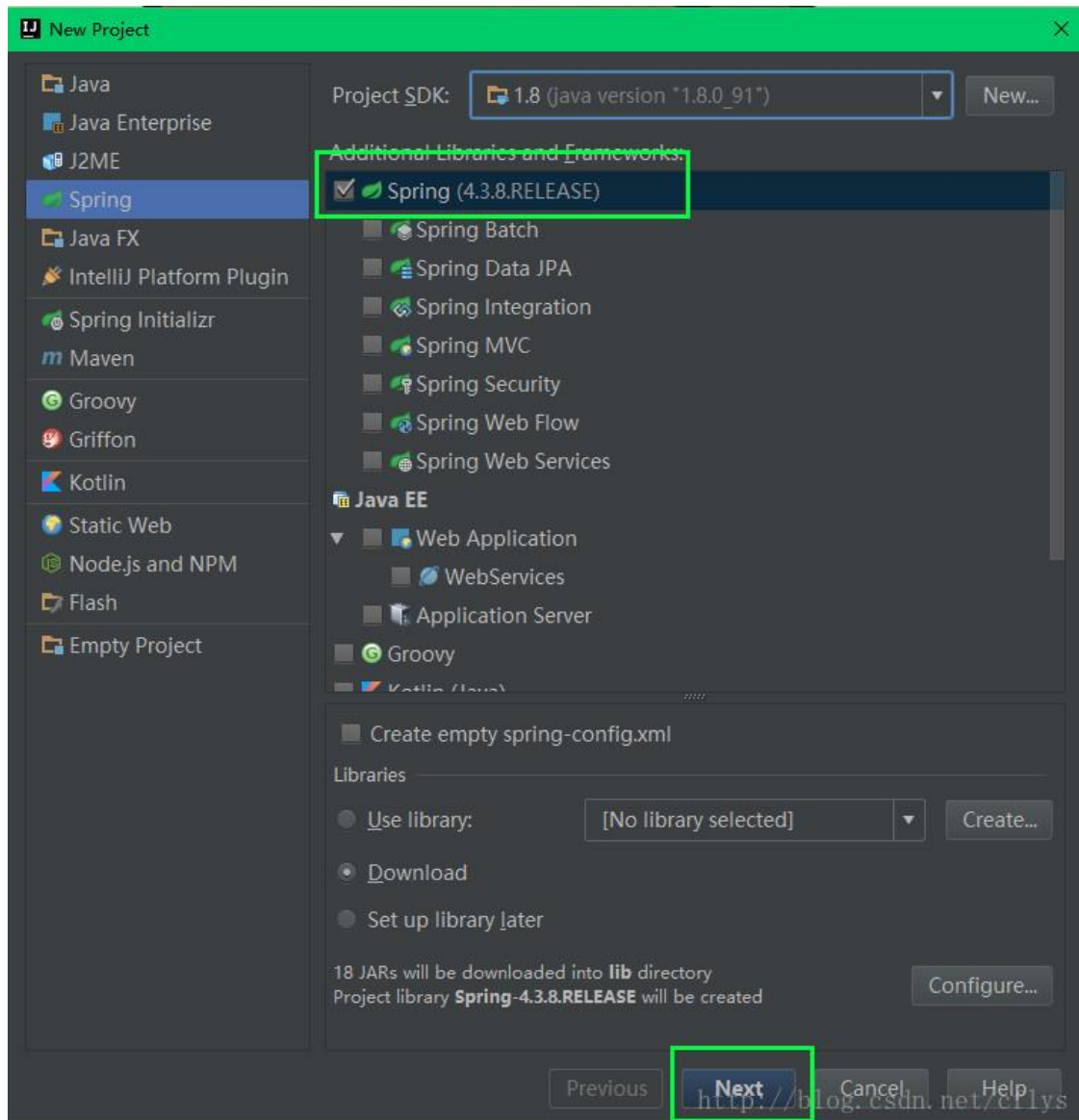
JVM: Java HotSpot™ 64-Bit Server VM by Oracle Corporation

在 IntelliJ IDEA 中创建 Spring 项目

在首先点击 Create New Project

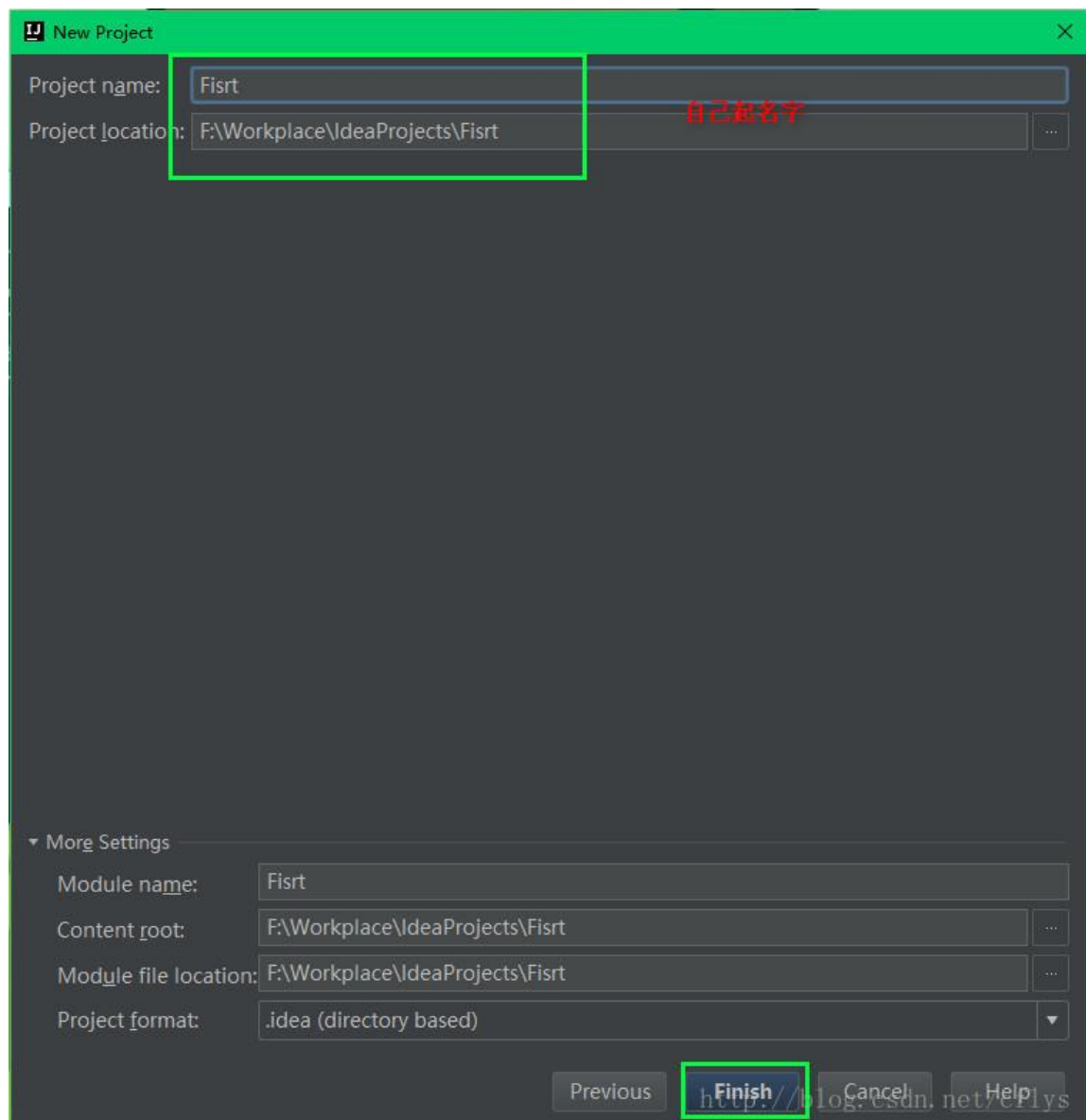


勾选 Spring 然后 next



注意：可以勾选 create empty spring-config.xml 自动生成 spring 配置文件，也可以生成项目后直接右键创建新的 xml 文件。

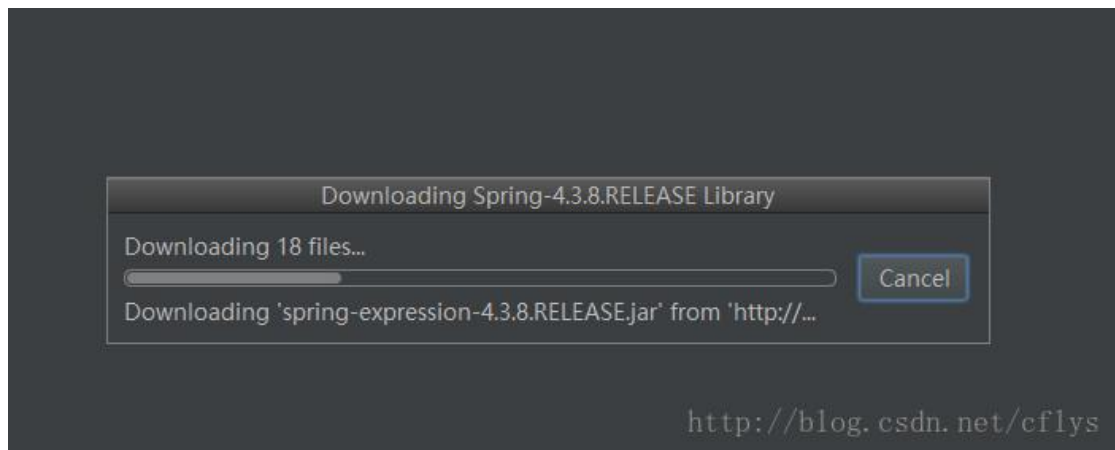
设置你项目所想要存放的路径以及名字



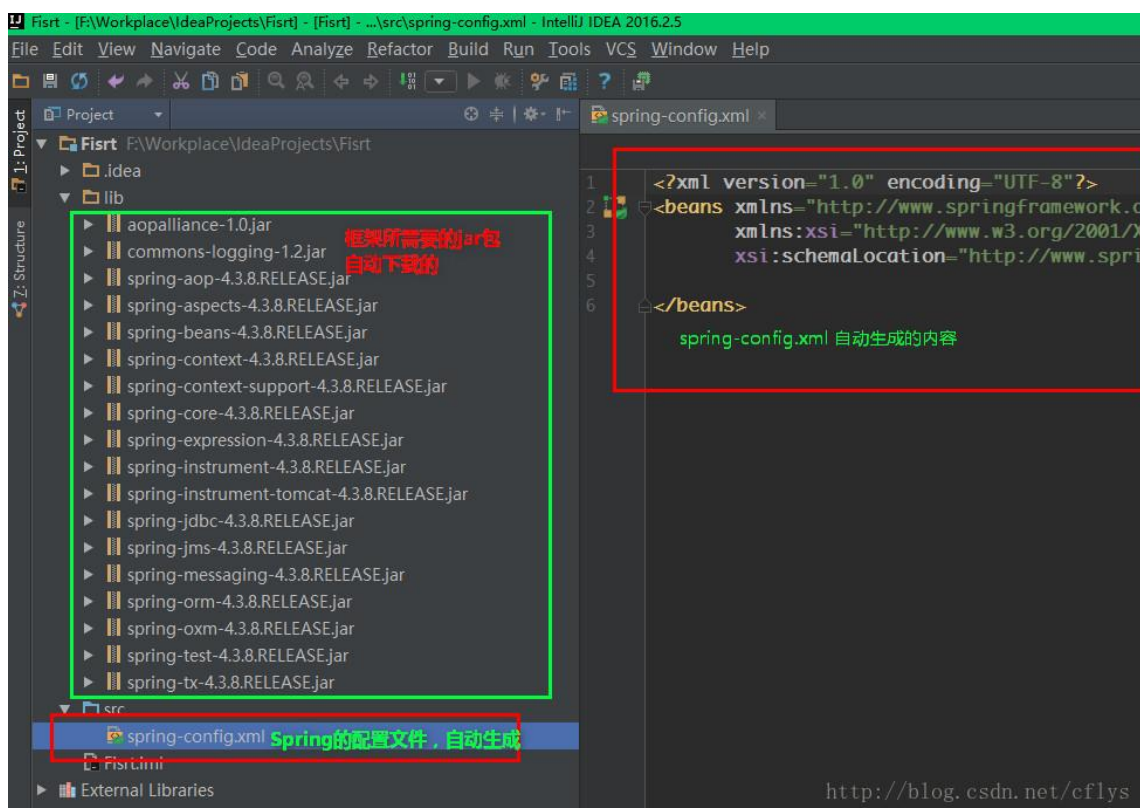
注：IntelliJ IDEA 中的 **project** 相当于 eclipse 中的 **workspace**，而 **module** 才相当于一个 **project**，所以我们不需要创建 **workspace**，IntelliJ IDEA 默认也会给你创建一个 **module**。

PS：First 拼错了，不要在意哈。。。

这时候 IntelliJ IDEA 就会自动下载 Spring 所需要的 jars，只需要等待就好。



下载好后，Spring 的 jars 和配置文件都准备好了。



一个简单的例子介绍框架的作用

首先创建一个 HelloWorld 类, 有一个 name 属性, 还有一个 sayHello 的方法, 还有一个 setter 方法用来设置 name 属性。



```
1  /** ... */
10
11  public class HelloWorld {
12      private String name;
13
14      public void setName(String name) {
15          this.name = name;
16      }
17
18      public void sayHello() {
19          System.out.println("Hello " + name);
20      }
21  }
```

<http://blog.csdn.net/cflys>

在我们不使用框架的时候，也就是平常的编程中，我们要调用 `sayHello` 这个方法，可以分为 3 步。

创建一个 `HelloWorld` 的实例对象

设置实例对象的 `name` 属性

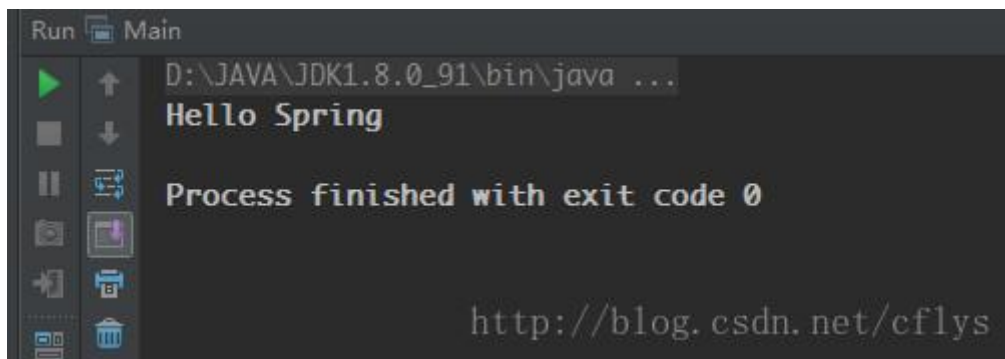
调用对象的 `sayHello()` 方法



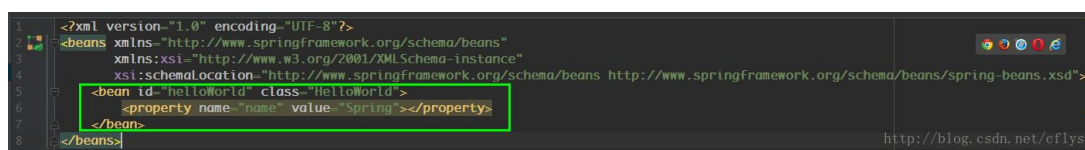
```
1  /** ... */
10  public class Main {
11      public static void main(String[] args) {
12          //不使用框架之前的步骤
13          //1. 创建一个HelloWrold对象
14          HelloWorld helloWorld = new HelloWorld();
15          //2. 为实例对象的属性赋值
16          helloWorld.setName("Spring");
17          //3. 调用对象的方法
18          helloWorld.sayHello();
19      }
20  }
```

<http://blog.csdn.net/cflys>

接着我们运行一下这个程序，成功输出了 `Hello Spring`。



接下来我们就要使用 Spring 了，首先在 Spring 的配置文件中加入如下内容。



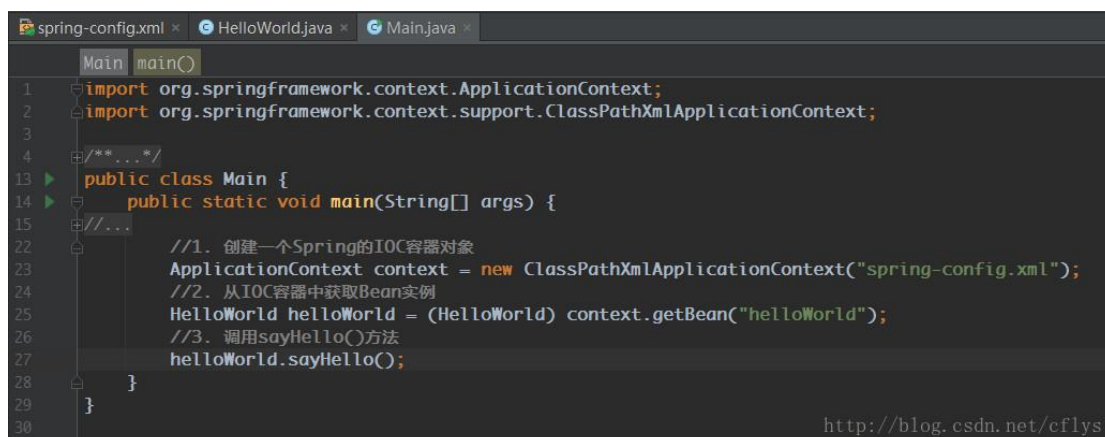
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="helloWorld" class="HelloWorld">
        <property name="name" value="Spring"></property>
    </bean>
</beans>
```

这时候我们就配置好了 HelloWorld Bean 的信息，我们再调用 sayHello()方法的时候就不向之前一样了，也需要 3 个步骤。

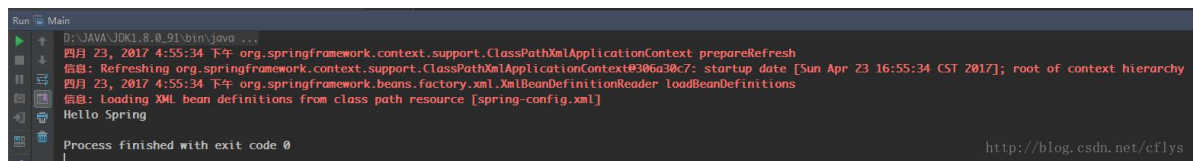
创建一个 Spring 的 IOC 容器对象

从 IOC 容器中获取 Bean 实例

调用 sayHello()方法



这么写好像和自己之前编程的时候不一样啊，可以运行结果吗，我们直接试一下就好。



```
Run - Main
D:\JAVA\JDK1.8.0.91\bin\java ...
四月 23, 2017 4:55:34 下午 org.springframework.context.support.ClassPathXmlApplicationContext prepareRefresh
信息: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@206a30c7: startup date [Sun Apr 23 16:55:34 CST 2017]; root of context hierarchy
四月 23, 2017 4:55:34 下午 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
信息: Loading XML bean definitions from class path resource [spring-config.xml]
Hello Spring
Process finished with exit code 0
http://blog.csdn.net/cflys
```

好像不太一样啊，输出了我们想要的 **Hello Spring**，但是好像多了许多其他的東西啊。这些其实是 **Spring** 输出的日志而已。

第一次使用 **Spring**，我们明明没有创建 **HelloWorld** 的实例对象，只是配置了下 **Spring** 的配置文件，怎么就能得出正确的结果呢，这是因为我们使用了 **Spring** 的 **IOC** 功能，把对象的创建和管理的功能都交给了 **Spring** 去管理，我们需要对象的时候再和 **Spring** 去要就行。

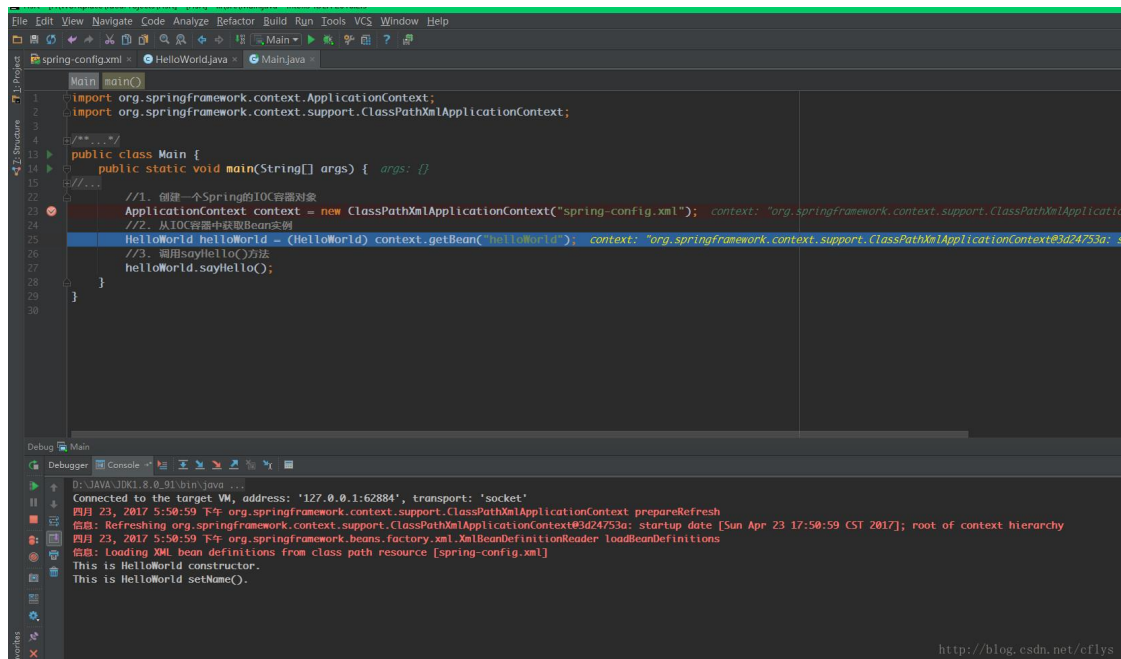
那么什么时候 **new** 的对象呢

我也不知道，哈哈，所以首先修改一下 **HelloWorld** 类的构造方法和 **setter** 方法。



```
1  /**
10
11 public class HelloWorld {
12     private String name;
13
14     public HelloWorld(){
15         System.out.println("This is HelloWorld constructor.");
16     }
17
18     public void setName(String name) {
19         System.out.println("This is HelloWorld setName().");
20         this.name = name;
21     }
22
23     public void sayHello() {
24         System.out.println("Hello " + name);
25     }
26 }
27
http://blog.csdn.net/cflys
```

然后直接添加断点进入 **Debug** 模式。



我们可以看到，当执行到第一步创建 IOC 容器对象的时候就调用了 HelloWorld 类的构造方法和 setter 方法。

对 Spring 的作用的初步了解

从上面的例子可以得出结论，Spring 到底帮我们做了什么，当我们没有使用 Spring 的时候，调用 sayHello()方法需要 3 步：

创建一个 HelloWorld 的实例对象

设置实例对象的 name 属性

调用对象的 sayHello()方法

后来也需要 3 步：

创建一个 Spring 的 IOC 容器对象

从 IOC 容器中获取 Bean 实例

调用 sayHello()方法

然后我们探究了什么时候 new 的 helloWorld 对象，我们可以看出，Spring 帮我们完成了前 2 步，也就是创建实例对象以及设置对象的属性，也就是说我们可以把对象的创建和管理工作交给 Spring 去完成，不需要自己去 new 对象，也不要设置对象的属性，只要写好 Spring 的配置文件，Spring 就可以帮我们去做，当我们需要对象的时候，直接去找 Spring 去要就行。

作者：cflys

来源：CSDN

原文：<https://blog.csdn.net/cflys/article/details/70598903>

版权声明：本文为博主原创文章，转载请附上博文链接！