# Spring的配置

## 基本步骤

1. Jar包准备：spring-framwork相关(4.3.5.RELEASE)，commons-logging(1.2)
2. 在src源目录下设置配置文件applicationContext.xml，相关配置见网上。
3. 在使用bean注入的时候，要事先加载配置文件

ApplicationContext ac=

new ClassPathXmlApplicationContext("applicationContext.xml");

//根据绝对路径加载xml文件

## Jar包简介

(1) spring-core.jar

**包含Spring框架基本的核心工具类**，Spring其它组件要都要使用到这个包里的类，是其它组件的基本核心，当然你也可以在自己的应用系统中使用这些工具类。

(2) spring-beans.jar

所有应用都要用到的，**包含访问配置文件、创建和管理bean以及进行Inversion of Control / Dependency Injection（IoC/DI）操作相关的所有类**。

如果应用只需基本的IoC/DI支持，引入spring-core.jar及spring-beans.jar文件就可以了。

(3) spring-aop.jar

包含在应用中**使用Spring的AOP特性时所需的类**。使用基于AOP的Spring特性，如**声明型事务管理**（Declarative Transaction Management），也要在应用里包含这个jar包。

(4) spring-context.jar

为Spring核心提供了大量扩展。可以找到使用Spring ApplicationContext特性时所需的全部类，JDNI所需的全部类，

UI方面的用来与模板（Templating）引擎如Velocity、FreeMarker、JasperReports集成的类，以及校验Validation方面的相关类。

(5) spring-dao.jar

**包含Spring DAO、Spring Transaction进行数据访问的所有类**。为了使用声明型事务支持，还需在自己的应用里包含spring-aop.jar。

(6) spring-hibernate.jar

　　包含Spring对Hibernate 2及Hibernate 3进行封装的所有类。

(7) spring-jdbc.jar

　　包含对Spring对JDBC数据访问进行封装的所有类。

(8) spring-orm.jar

**包含Spring对DAO特性集进行了扩展，使其支持 iBATIS、JDO、OJB、TopLink**，因为Hibernate已经独立成包了，现在不包含在这个包里了。

大部分的类都要依赖spring-dao.jar里的类，用这个包时需要同时包含spring-dao.jar包。

(9) spring-remoting.jar

包含支持EJB、JMS、远程调用Remoting（RMI、Hessian、Burlap、Http Invoker、JAX-RPC）方面的类。

(10) spring-support.jar

包含支持缓存Cache（ehcache）、JCA、JMX、邮件服务（Java Mail、COS Mail）、任务计划Scheduling（Timer、Quartz）方面的类。

(11) spring-web.jar

包含Web应用开发时，用到Spring框架时所需的核心类，包括自动载入WebApplicationContext特性的类、Struts与JSF集成类、文件上传的支持类、Filter类和大量工具辅助类。

(12) spring-webmvc.jar

包含Spring MVC框架相关的所有类。包含国际化、标签、Theme、视图展现的FreeMarker、JasperReports、Tiles、Velocity、XSLT相关类。

当然，如果你的应用使用了独立的MVC框架，则无需这个JAR文件里的任何类。

(13) spring-mock.jar

包含Spring一整套mock类来辅助应用的测试。Spring测试套件使用了其中大量mock类，这样测试就更加简单。

模拟HttpServletRequest和HttpServletResponse类在Web应用单元测试是很方便的。

## 注意

1. 导入Jar包要spring-core和common公共包，版本要4.3.5和1.2以上。
2. Maven的依赖



1. commons-logging包作用是显示spring程序运行时的程序日志显示，以及self日志门面接口。

# 二、IOC的基本步骤

1. SpringIOC（控制反转）：把对象的创建，初始化，销毁等工作交给Spring容器来做。由Spring容器控制对象的生命周期。

2. 步骤

(1) 启动Spring容器；

(2) 从Spring容器中把对象取出来；

(3) 对象调用实际的业务方法。

* 1. 启动Spring容器：

1. 在类路径下寻找配置文件来**实例化容器**：ApplicationContext context = new ClassPathXmlApplicationContext("配置文件类路径");

通过这种方式加载。需要将spring配置文件放到当前项目的classpath路径下。classpath路径指的是当前项目的src目录，该目录是java源文件的存放位置。

1. 在文件系统路径下寻找配置文件来实例化容器：ApplicationContext context = new FileSystemXmlApplicationContext("配置文件系统路径");
2. 注：经常采用第一种方式启动Spring容器。

2.2 从Spring容器中把对象取出来：

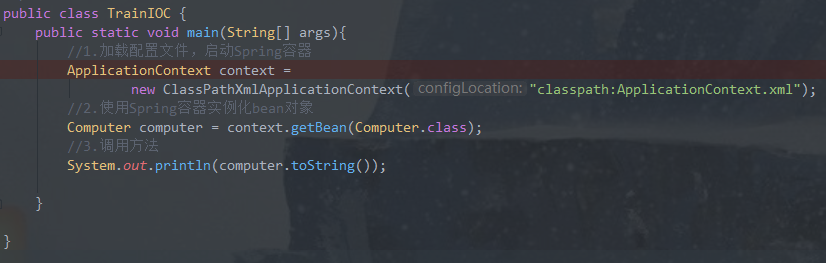
(1) context.getBean("配置文件中bean的id"); 如：

HelloWorld helloWorld = (HelloWorld)context.getBean("helloWorld");

Person person = context.getBean(“person”,Person.Class);

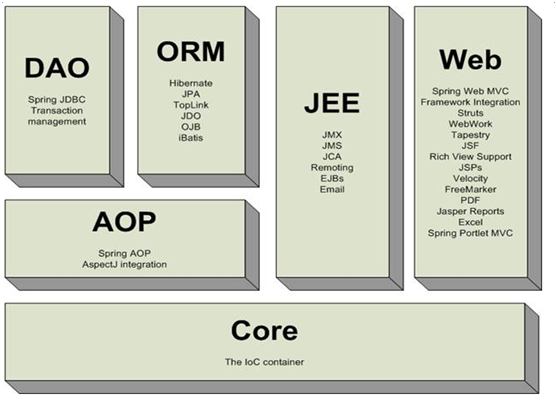
2.3 对象调用实际的业务方法:

helloWorld.hello();



Spring IOC、AOP、事务管理、事件机制

# 三、Spring基本框架结构



1. Spring DAO：Spring提供了对JDBC的操作支持：JdbcTemplate模板工具类 。
2. Spring ORM：Spring可以与ORM框架整合。例如Spring整合Hibernate框架，其中Spring还提供HibernateDaoSupport工具类，简化了Hibernate的操作 。
3. Spring WEB：Spring提供了对Struts、Springmvc的支持，支持WEB开发。与此同时Spring自身也提供了基于MVC的解决方案 。
4. Spring AOP：Spring提供面向切面的编程，可以给某一层提供事务管理，例如在Service层添加事物控制 。
5. Spring JEE：J2EE开发规范的支持，例如EJB 。
6. Spring Core：提供IOC容器对象的创建和处理依赖对象关系 。