# Spring

http://www.cnblogs.com/yuanfuqiang/p/5834496.html

1.interface BeanFactory{

让bean工厂持有一个Map<String,BeanDefinition>

//根据bean的名字，获取在IOC容器中得到bean实例

Object getBean(String name) throws BeansException;

10     //根据bean的名字和Class类型来得到bean实例，增加了类型安全验证机制。

11      Object getBean(String name, Class requiredType) throws BeansException;

}

1. 初始化bean工厂呢，或者说建立IOC容器。

      那么从现在来看，我们需要什么才能把Map填充呢？也就是初始化bean工厂呢，或者说建立IOC容器。

             1.需要一个File指向我们的XML文件（本文的配置文件都已XML为例，因为这是我们最熟悉的），专业点可以叫资源定位，简单点可以说我们需要一些工具来完成找到XML文件的所在位置。

             2.需要一个Reader来读取我们的XML文件，专业点叫DOM解析，简单点说，就是把XML文件的各种定义都给拿出来。

             3.需要将读出来的数据都设置到Map当中。

             这三步总结起来就是定位、解析、注册。

# Springmvc

# Mybatis

# Ioc理解

A:n IoC容器

简单的理解就是：实现IoC思想，并提供对象创建、对象装配以及对象生命

周期管理的软件就是IoC容器。

B:n IoC理解

1：应用程序无需主动new对象；而是描述对象应该如何被创建即可

IoC容器帮你创建，即被动实例化；

2：应用程序不需要主动装配对象之间的依赖关系，而是描述需要哪个服务

IoC容器会帮你装配（即负责将它们关联在一起），被动接受装配；

3：应用程序主动变被动，（不取，等待容器主动注入）（实现了松散耦合）体现好莱坞法则：别打电话给我们，我们会打给你

3：ioc和di是一个概念？是对同一个问题的不同角度描述。

控制反转：从容器角度说。容器反过来控制应用程序，控制应用程序所需要的外部资源

依赖注入：从应用程序的角度说。应用程序依赖于容器，依赖他为注入所需要的资源

# Aop理解

解决横切关注点（我们关注的多个公共模块）的模块化的编程范式

Aspect Oriented Programming)（面向切面编程

AOP是一种编程范式，提供从另一个角度来考虑程序结构以完善面向对象编

程（OOP）。

AOP为开发者提供了一种描述横切关注点的机制，并能够自动将横切关注点

织入到面向对象的软件系统中，从而实现了横切关注点的模块化。

AOP能够将那些与业务无关，却为业务模块所共同调用的逻辑或责任，例

如事务处理、日志管理、权限控制等，封装起来，便于减少系统的重复代码，降

低模块间的耦合度，并有利于未来的可操作性和可维护性。

## n AOP能干什么，

也是AOP带来的好处

1：降低模块的耦合度

2：使系统容易扩展

3：设计决定的迟绑定：使用AOP,设计师可以推迟为将来的需求作决定，因为它

可以把这种需求作为独立的方面很容易的实现。

4：更好的代码复用性