SSM框架和Redis

1. Spring框架

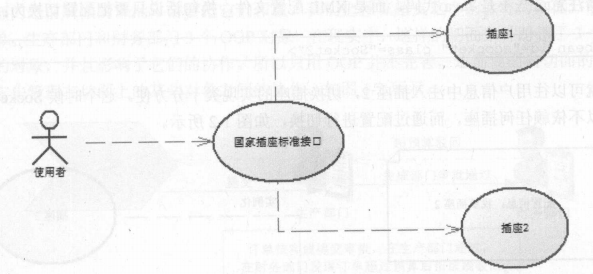
Spring是Java应用最广的框架，它的理念包括IoC和AOP

1. Spring IoC简介（IoC的目标是为了管理Bean，而Bean是面向对象（OOP）的基础

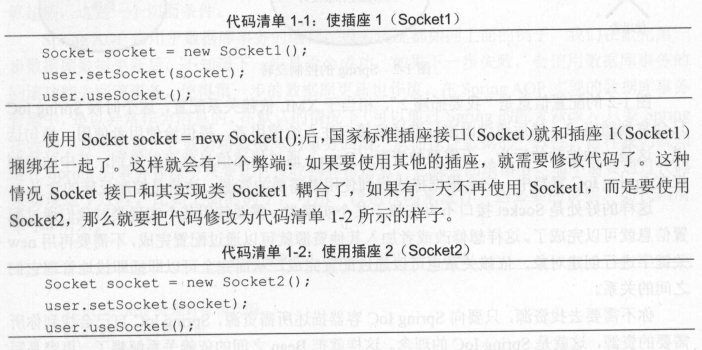
设计）

IoC是一个容器，它认为一切Java资源都是Java Bean，容器的目标就是管理这些Bean和它们之间的关系，所以在Spring IoC里面装载了各种Bean,也可以理解为Java的各种资源。

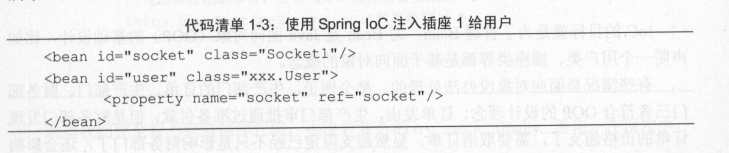
比如插座，他依赖国家标准（这个标准可以定义为一个接口（Socket）），现在有两种插座（Socket1和Socket2）



如果采用之前的方式，当出现对象数量很多的情况下，修改源码对系统的可靠性会是极大的挑战，而Spring IoC可以解决这个问题



Spring IoC可以不用new的方式创建对象，而是采用配置文件xml的方式，然后让Spring IoC自己通过配置去找插座，先用一段XML描述插座和用户的引用插座1



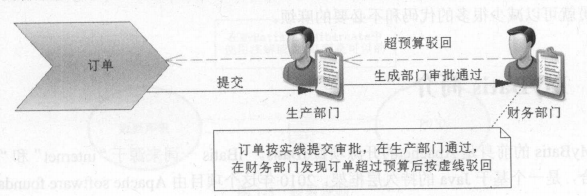
如果切换为插座2，则将XML配置文件配置切换为

<bean id=”socket” class=”Socket2”/>,这个时候Spring IoC只会拿到插座2，然后通过国家标准插座接口注入给使用者，提供给使用者使用。这是一种被动的行为，而需要的资源（Bean）通过描述信息就可以得到，其中控制反转权在Spring IoC容器中，它会根据描述找到使用者需要的资源，这就是控制反转的含义。

IoC的好处在于Socket接口不再依赖于某个实现类，需要使用某个实现类时我们通过配置文件就可以完成了。这样想修改或者加入其他资源可以通过配置完成，不再需要new关键字进行创建对象，依赖关系也可以通过配置完成，从而完全可以即插即拔地管理它们之间的关系。**你不需要去找资源，只要向Spring IoC容器描述所需的资源，Spring IoC自己会找到你需要的资源，这就是Spring IoC的理念。这样就把Bean之间的依赖关系解耦了，更容易写出结构清晰的程序。**

1. Spring AOP

有些情况是面向对象没有办法处理的



Spring AOP常用于数据库事务的编程，是以异常作为消息的。就是说，**做完第一步数据库数据更新后，不知道下一步是否会成功，如果下一步失败，是使用数据库的回滚功能去回滚事务，使得第一步的数据库更新也作废。**只要Spring接收到了异常消息，它就会将数据库的事务回滚，从而保证数据的一致性，而不需要代码来实现这个过程。

**Spring AOP的好处：**在Spring中我们完全看不到数据库代码，也没有复杂的try…catch…finally，这些东西都被Spring屏蔽了，不需要关注，只需要关注业务代码，知道发生了异常，Spring会回滚事务就够了。

1. MyBatis简介

前身是Apache的开源项目iBatis，是基于Java的持久层框架。MyBatis的优势在于灵活，它几乎可以代替JDBC，同时提供了接口编程。Mybatis的数据库访问层DAO是不需要实现类的，只需要一个接口和XML（或者注解）

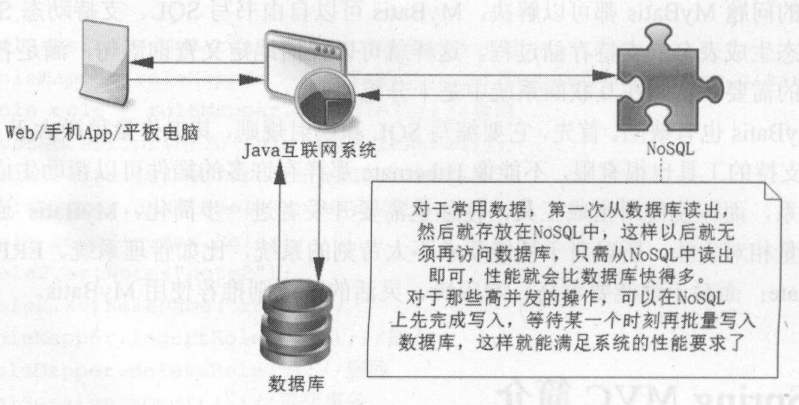
Mybatis提供自动映射、动态SQL、级联、缓存、注解、代码和SQL分离等特性，使用方便，同时也可以对SQL进行优化。因为其具有封装少、映射多样化、支持存储过程、可以进行SQL优化等特点，使他取代了Hibernate成为Java互联网中首选的持久化框架。

1. Mybatis和Hibernate的异同
   1. 都是持久层框架，都涉及数据库，都是依靠某种方法将数据库的表和POJO映射起来
   2. 把POJI对象和数据库表相互映射的框架称为对象关系映射（Object Relational Mapping，ORM）框架，无论是Mybatis或者Hibernate都可以称为ORM框架，只是Hibernate的设计理念是完全面向POJO的，而Mybatis不是。Hibernate基本不需要再编写SQL就可以通过映射关系操作数据库，是一种全表映射的体现；而Mybatis不同，它需要我们提供SQL去运行
   3. 在移动互联网时代，Mybatis称为目前互联网Java持久化框架的首选，Mybatis不屏蔽SQL，这样的优势在于，程序员可以自己制定SQL规则，无需Hibernate自动生成规则，这样能够更精确地定义SQL,从而优化性能，它更符合移动互联网高并发、大数据、高性能、高响应的要求
   4. Mybatis需要提供接口和SQL，这意味着它的工作量比Hibernate大，但是由于自定义SQL、映射关系，所以其灵活性、可优化性就超过了Hibernate，（互联网的可优化性、灵活性是非常重要的，因为一条SQL的性能可能相差十几倍到几十倍，这对于互联网是非常重要的）
2. Spring MVC简介

Spring MVC结构层次清晰，类比较简单，并且与Spring的核心IoC和AOP无缝对接，成为互联网时代的主流框架。MVC把应用程序分成不同的方面，同时提供这些元素之间的松耦合。Model（模型），封装了应用程序的数据和它们组成的POJO；View（视图），负责把模型数据渲染到视图上，将数据以一定的形式展现给用户，Controller（控制器），负责处理用户请求，并建立适当的模型把它传递给视图渲染。

Spring MVC中的重点在于它的流程和一些重要的注解，包括控制器，视图解析器、视图等重要的内容

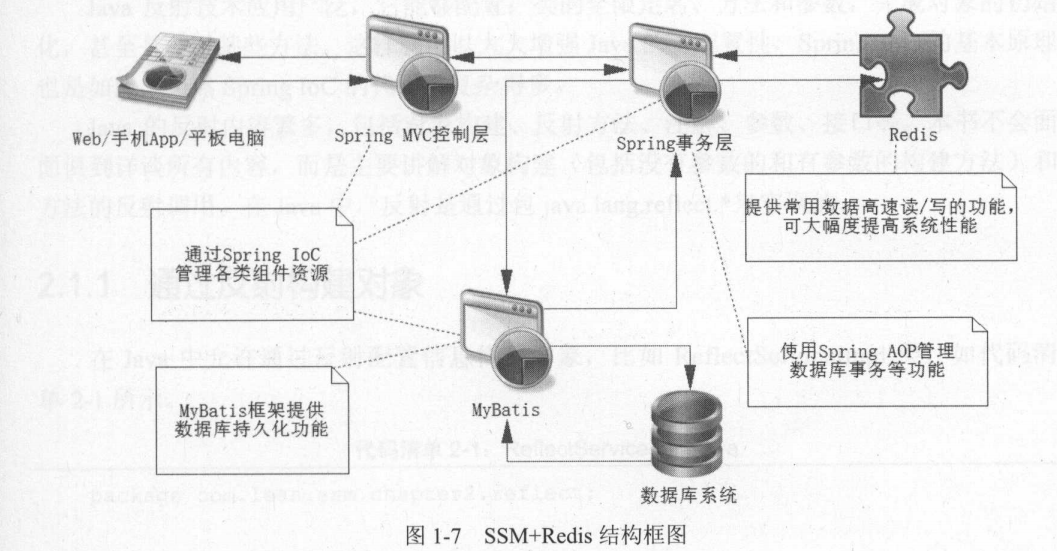
1. 最流行的NoSQL-----Redis



Redis在互联网中的作用很大，可以很大程度上提高互联网系统的性能。它具备持久化功能，也可以作为一种缓存工具。它的读入速度很快，适合读多写少；作为缓存，它可以支持大数据存入内存中，只要命中率高，它就能快速响应，因为在内存中的数据读/写比数据库的读/写磁盘的速度快几十倍或上百倍。

Redis已经成为主要的NoSQL工具，原因如下：

1. 响应速度快，Redis响应速度非常快，每秒可以执行大约110000写入操作，或者81000个读操作，速度远超数据库。如果存入一些常用的数据，就能有效提高系统性能
2. 支持5中数据类型
3. 操作都是原子性的
4. SSM+Redis结构框图及描述



Spring IoC承担了一个资源管理、整合、即插即拔的功能

Spring AOP可以提供切面管理，特别是数据库事务管理的功能

Spring MVC用于把模型、视图和控制器分层，组合成一个有机灵活的系统

MyBatis提供一个数据库访问的持久层，通过MuBatis-Spring项目，便能和Spring无缝对接

Redis作为缓存工具，提供了一个高速度处理数据和缓存数据的功能，使得系统大部分数据只需要访问缓存，再无需从数据库磁盘中重复读/写；在一些需要高速运算的场合中，也可以用它来完成运算，再把数据批量存入数据库，这样极大的提升了互联网系统的性能和响应能力。