1.Web.xml配置内容

配置过滤器filter(比如编码过滤器)和监听器listener(统计网站在线人数等;监听session创建和销毁，监听对象属性的增加和删除)，配置errorpage

load-on-startup

load-on-startup 元素在web应用启动的时候指定了servlet被加载的顺序，它的值必须是一个整数。如果它的值是一个负整数或是这个元素不存在，那么容器会在该 servlet被调用的时候，加载这个servlet 。如果值是正整数或零，容器在配置的时候就加载并初始化这个servlet，容器必须保证值小的先被加载。如果值相等，容器可以自动选择先加载谁

context-param，webapp的环境参数，它用于向 ServletContext 提供键值对

每个应用都会有一个[**ServletContext**](http://baike.baidu.com/view/3923348.htm)对象与之关联

加载顺序ServletContext -> listener -> filter -> servlet

与在web.xml文件中的顺序无关

如果不用struts2框架，web.xml可以用于配置servlet和servletmapping(servlet映射)

配置session超时时间sessionconfig

指定欢迎页面<welcome-file-list>

Taglib，设置jsp用到的tag library路径

2.hibernate-config.xml(spring-mvc)

配置session-factory ：url信息、数据库方言、数据库驱动信息

配置事务管理 transaction-manager

3.applicationContxt.xml(SSH)

配置session-factory和datasource 配置事务管理

配置要扫描的有注解的文件 base-package 包路径

4.springmvc.xml

配置视图解析器 将modelandview及字符串解析为具体的页面

什么是Spring

Spring 是个java企业级应用的开源开发框架。Spring主要用来开发Java应用，但是有些扩展是针对构建J2EE平台的web应用。Spring 框架目标是简化Java企业级应用开发

Spring的好处

* **轻量：**Spring 是轻量的，基本的版本大约2MB。
* **控制反转：**Spring通过控制反转实现了松散耦合，对象们给出它们的依赖，而不是创建或查找依赖的对象们。Spring通过一种称作控制反转（[IoC](http://baike.baidu.com/subview/146665/146665.htm)）的技术促进了低耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。
* **面向切面的编程(AOP)：**Spring支持面向切面的编程，并且把应用业务逻辑和系统服务分开。允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和[事务](http://baike.baidu.com/view/121511.htm)（[transaction](http://baike.baidu.com/view/2368136.htm)）管理）进行[内聚性](http://baike.baidu.com/view/2424328.htm)的开发。[应用对象](http://baike.baidu.com/view/5314991.htm)只实现它们应该做的——完成业务逻辑——仅此而已
* **容器：**Spring 包含并管理应用中对象的生命周期和配置。
* **MVC框架**：Spring的WEB框架是个精心设计的框架，是Web框架的一个很好的替代品。
* **事务管理：**Spring 提供一个持续的事务管理接口，可以扩展到上至本地事务下至全局事务（JTA）。
* **异常处理：**Spring 提供方便的API把具体技术相关的异常（比如由JDBC，Hibernate or JDO抛出的）转化为一致的unchecked 异常

**什么是Spring IOC 容器？**

Spring IOC 负责创建对象，管理对象（通过依赖注入（DI），装配对象，配置对象，并且管理这些对象的整个生命周期。

Spring的MVC框架是围绕DispatcherServlet来设计的，它用来处理所有的HTTP请求和响应。

**Spring MVC工作原理**  
1. 客户端请求提交到DispatcherServlet  
2. 由DispatcherServlet控制器查询一个或多个HandlerMapping，找到处理请求的Controller  
3. DispatcherServlet将请求提交到Controller  
4. Controller调用业务逻辑处理后，返回ModelAndView  
5. DispatcherServlet查询一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图  
6. 视图负责将结果显示到客户端

**Spring的事务管理**   
事务就是对一系列的数据库操作（比如插入多条数据）进行统一的提交或回滚操作，如果插入成功，那么一起成功，如果中间有一条出现异常，那么回滚之前的所有操作。这样可以防止出现脏数据，防止数据库数据出现问题。

AOP，面向切面编程，就是把可重用的功能提取出来，然后将这些通用功能在合适的时候织入到应用程序中，比如事务管理、权限控制、日志记录、性能统计等

. Hibernate3 提供了属性的延迟加载功能

当Hibernate在查询数据的时候，数据并没有存在与内存中，当程序真正对数据的操作时，对象才存在与内存中，就实现了延迟加载，他节省了服务器的内存开销，从而提高了服务器的性能

如何优化Hibernate？  
1.使用双向一对多关联，不使用单向一对多  
2.灵活使用单向一对多关联  
3.不用一对一，用多对一取代  
4.配置对象缓存，不使用集合缓存  
5.一对多集合使用Bag,多对多集合使用Set  
6. 继承类使用显式多态  
7. 表字段要少，表关联不要怕多，有二级缓存撑腰

Struts的工作流程:  
在web应用启动时就会加载初始化ActionServlet,ActionServlet从struts-config.xml文件中读取配置信息,把它们存放到各种配置对象  
当ActionServlet接收到一个客户请求时,将执行如下流程.  
-(1)检索和用户请求匹配的ActionMapping实例,如果不存在,就返回请求路径无效信息;   
-(2)如果ActionForm实例不存在,就创建一个ActionForm对象,把客户提交的表单数据保存到ActionForm对象中;   
-(3)根据配置信息决定是否需要表单验证.如果需要验证,就调用ActionForm的validate()方法;   
-(4)如果ActionForm的validate()方法返回null或返回一个不包含ActionMessage的ActuibErrors对象, 就表示表单验证成功;   
-(5)ActionServlet根据ActionMapping所包含的映射信息决定将请求转发给哪个Action,如果相应的 Action实例不存在,就先创建这个实例,然后调用Action的execute()方法;   
-(6)Action的execute()方法返回一个ActionForward对象,ActionServlet在把客户请求转发给 ActionForward对象指向的JSP组件;   
-(7)ActionForward对象指向JSP组件生成动态网页,返回给客户;

load()和get()的区别

hibernate对于load方法认为该数据在数据库中一定存在，可以放心的使用代理来延迟加载，load默认支持延迟加载，在用到对象中的其他属性数据时才查询数据库，但是万一数据库中不存在该记录，只能抛异常ObjectNotFoundEcception；所说的load方法抛异常是指在使用该对象的数据时，数据库中不存在该数据时抛异常，而不是在创建这个对象时。由于session中的缓存对于hibernate来说是个相当廉价的资源，所以在 load时会先查一下session缓存看看该id对应的对象是否存在，不存在则创建代理（load时候之查询一级缓存，不存在则创建代理）。get() 现在一级缓存找，没有就去二级缓存找，没有就去数据库找，没有就返回null ；而对于get方法，hibernate一定要获取到真实的数据，否则返回null

- REQUIRES\_NEW总是创建新事务

- NOT\_SUPPORTED存在事务则挂起，一直执行非事务操作 - NEVER总是执行非事务，如果当前存在事务则抛异常 - NESTED嵌入式事务

9.Hibernate中的update()和saveOrUpdate()的区别.  摘自hibernate说明文档：

saveOrUpdate()做下面的事:

如果对象已经在本session中持久化了，不做任何事

如果另一个与本session关联的对象拥有相同的持久化标识(identifier)，抛出一个异常 如果对象没有持久化标识(identifier)属性，对其调用save()

如果对象的持久标识(identifier)表明其是一个新实例化的对象，对其调用save()

如果对象是附带版本信息的（通过<version>或<timestamp>）并且版本属性的值表明其是一个新实例化的对象，save()它。否则update() 这个对象

10.Spring对多种ORM框架提供了很好的支持，简单描述在Spring中使用Hibernate的方法，并结合事务管理。

getHiberanteTemplate里面提供了save，update，delete，find等方法。

简单说一个：如果配置了声明式事务，当执行getHibernateTemplate的各种方法的时候，事务会

自动被加载

如果没有配置事务，那么以上操作不会真正的被同步到数据库，除非配置了hibernate的 autocommit=true

spring的事务有几种方式？谈谈spring事务的隔离级别和传播行为。

spring事务分两种形式，声明式事务和编程式事务，spring提供了一个事务的接口

PaltformTractionManager接口，针对不同的事务，spring进行了不同的实现,对hibernate事务 的实现HIbernateTractionManager,对JDBC的JdbcTractionManager，

DataSourceTractionManager以及JdoTractionManager。接口platformTractionManager提供了三 个方法，获取事务，提交和回滚的方法。

Hibernate工作原理及为什么要用？

 原理：

1.读取并解析配置文件

2.读取并解析映射信息，创建SessionFactory

3.打开Sesssion

4.创建事务Transation

5.持久化操作

6.提交事务

7.关闭Session

8.关闭SesstionFactory

为什么要用：

1.    对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。 2.    Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作

3.    hibernate使用Java反射机制，而不是字节码增强程序来实现透明性。

4.    hibernate的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系

**什么是Spring beans?**

Spring beans 是那些形成Spring应用的主干的java对象。它们被Spring IOC容器初始化，装配，和管理。这些beans通过容器中配置的元数据创建。比如，以XML文件中<bean/> 的形式定义。

声明式事务管理，因为它对应用代码的影响最小，因此更符合一个无侵入的轻量级容器的思想。声明式事务管理要优于编程式事务管理，虽然比编程式事务管理（这种方式允许你通过代码控制事务）少了一点灵活性。

**切点pointcut**

切入点是一个或一组连接点，通知将在这些位置执行。可以通过表达式或匹配的方式指明切入点

**Advisor**

是Pointcut和Advice的配置器，它包括**Pointcut**和**Advice**，是将Advice注入程序中Pointcut位置的代码

**通知Advice**

它是某个连接点所采用的处理逻辑，也就是向连接点注入的代码