EJB 内容

EJB 的角色和三个对象

一个完整的基于 EJB 的分布式计算机构由 6 个角色组成，这 6 个角色可以由不同的开发商提供，这 6 个角色所实现的功能必须遵循 SUN 公司提供的 EJB 规范，这 6 个角色是：EJB 组件开发者，应用组合者，部署者，EJB 容器提供者，EJB 服务器提供者，系统管理员

三个对象：Remote(/Local)接口、HOME 接口，BEAN 类

EJB 容器提供的服务

1.生命周期管理

2.代码产生

3.持久性管理

4.安全事务管理

5.锁和并发行管理

EJB 规范中禁止的操作

1.不能操作线程和线程 API

2.不能操作 AWT

3.不能实现服务器功能

4.不能对静态属性存取

5.不能使用 IO 直接操作文件

6.不能加载本地库

7.不能使用 this 作为变量和返回

8.不能循环调用

Remote 接口和 HOME 接口的作用

Remote 接口定义了业务方法，用于 EJB 客户端调用 Remote 接口的业务方法

HOME 接口是 EJB 工厂，用于创建和移除查找 EJB 实例

Bean 实例的生命周期

Stateless Session Bean、Entity Bean、DriverMessage Bean 是存在缓冲池管理

Stateful Session Bean 是存在 Cache 管理

存在缓冲池管理的 Bean 在创建之后并不会从内存中清除，而是通过缓冲池调度机制不断的重用 EJB 实例

存在 Cache 管理的 Bean 通过使用激活和去激活机制保持 Bean 的状态，并且限制内存中实例的数量

EJB 的激活机制

Stateful Session Bean 举例：它的 Cache 大小决定可以同时存在的 Bean 实例的个数，根据 MRU 和 NRU 算法，Bean 的状态在激活和去激活状态之间迁徙。激活机制：当一个客户端调用一个实例执行实例的某个方法时，如果 EJB OBJECT 发现没有绑定相对应的实例，则从去激活 Bean 存储中激活此实例，状态变迁前会调用对应的 ejbActive 方法和 ejbpassivate 方法

EJB 的几种类型

Entity Bean 分为 Bean 管理的持续性和容器管理的持续性

Session Bean 分为 Stateless Session Bean 和 Stateful Session Bean

DriverMessage Bean

客服端调用 EJB 对象的几个基本步骤

首先设置 JNDI 服务工厂和 JNDI 服务地址系统属性，查找 HOME 接口，从 HOME 接口调用 create 方法创建 Remote 接口，通过 Remote 接口调用业务方法

1. 什么是 EJB？

   EJB 的全称是 Enterprise javabeans。是 JAVA 中的商业应用组件技术。EJB 结构中的角色 EJB 组件结构是基于组件的分布式计算结构，是分布式应用系统中的组件。

   EJB 是一种技术，而不是一种产品。

   EJB 是一种标,准描述了构建应用组件要解决的可扩展性、分布式、事务处理、数据存储、安全性。

2. 为什么选择 EJB？

   EJB 容器完成繁杂的的工作，应用开发人员只需要关注业务逻辑的实现，而不必要关注底层的实现机制，支持事务处理，安全性，可扩展性。

3. EJB 服务器的功能？

   管理 EJB 容器、提供对操作系统服务的存取、提供 JAVA 相关的服务（通过 JNDI 访问命名空间，基于 OTS 的事务服务）。

4. EJB 分类

   EJB 分为 Entity Bean、Session Bean、DriverMessage Bean

Session Bean :Stateful Session Bean、 Stateless Session Bean

5. 什么是会话 Bean?

   表示业务进程的一种 EJB，用于和客户端进行会话

   特点：代表客户端进行操作、可以识别事务处理、存在时间短，在客户端结束会话时结束。

6. JNDI

   JNDI：JNDI(Java Naming and Directory Interface，Java 命名和目录接口

7. @Remote、@Local

   @Remote 表示一个接口是远程接口

   @Local 表示一个接口是本地接口

8. @Stateless、@Stateful

   @Stateless 表示一个无状态的 Session Bean

   @Stateful 表示一个有状态的 Session Bean

9. Context 上下文环境

   Context context = new InitialContext();工程环境变量中必须要有一个 JNDI.properties 文件

   访问 EJB：context.lookUp("SessionBean Name/Remote");

   若无 JNDI.properties 文件则需要手写上下文编码

   Properties pop = new Properties;

   Pop.put();

   Context context = new InitialContext(pop);

10.其他

有状态 Session Bean 必须要实现 serializable

这样 EJB 容器才能在他们不再使用时序列化存储他们的状态信息

webService 内容

1.什么是 WebService？

WebService 是使原本各孤立之间的站点的数据能够相互通讯、共享而提供的一种接口。它是基于网络的、分布式的模块化组件。

WebService 使用的协议是 Internet 上统一、开放的标准。如 HTTP,XML,WSDL 等。所以 WebService 可以在任何支持这些标准的环境下使用

2.什么是 SOAP 协议？

SOAP 协议（Simple Object Access Protocal 简单对象访问协议）是用于分散和分布式环境下网络信息交换的基于 XML 的通讯协议，软件组件和应用程序能够通过标准的 HTTP 协议进行通讯。用于交换 XML 编码的轻量级协议

它的设计目标是简单性和可扩展性，这样有助于大量异构程序和平台的互操作性，从而使存在的应用程序能够被广泛的用户所访问到。

3.WebService 的优缺点？

优点：

1.跨平台

2.成本小

3.SOAP 使用的是业界的 HTTP，XML 这些标准，得到所有重要公司的支持

4.数据是以 ASCLL 文本方式传输，调用方便，由于使用这种方式，它的数据能够很容易的通过防火墙，而不需要防火墙为程序单独开设一个“漏洞”

5.WebService 实现的技术难度小

6.在 C/S 程序中可以实现无整体刷新的与服务器打叫道读取数据

缺点：

1.WebService 使用了 XML 对数据封装，存在大量的数据在网络中传输

2.WebService 规范没有规定任何与实现相关的细节

4.JAXP

JAXP 提供了 JAVA 中使用 DOM,SAX,XSTL 的通用接口，在你的程序中你只要调用通用的接口，当你需要改变具体实现的时候也不需要修改代码

5.JAXM

为 SOAP 通讯提供访问方法和传输机制的 API

6.WSDL

WSDL 是一种 XML，用它来描述服务的一组端点，它对包含面向文档信息和面向过程信息的消息进行操作，首先对操作和消息进行抽象描述，并把这些描述绑定到具体的网络协议和消息格式上以

定义端点，相关的具体端点即组合成抽象端点（服务）

三大框架

1.struts 优缺点

Struts 是采用 Java Servlet/JavaServer Pages 技术的，开发 WEB 应用程序的开放源码的框架。Struts 具有组件的模块化、灵活性和重用性的优点。

优点：1.开源：是开发者能够深入了解它的内部实现机制

2.Taglib：具有丰富的标签灵活动用，能够提高开发效率

3.页面导航：使系统脉络更加清晰，通过一个 struts-config.xml 配置文件即可把握整个系统各部分之间的关系。

缺点：1.大量的使用标签，对于初学者来说有一定难度。

2.将系统分为 M、V、C 三个部分，在获得清晰的结构的同时，也增加了系统的复杂性

3.ActionForm 使用不便，无法进行单元测试

2.hibernate 优缺点

hibernate 是一个开放源码的关系映射框架，对 JDBC 进行了轻量级的封装，是 JAVA 程序员能随心所欲的使用面向对象编程操作数据库。

优点：1.hibernate 使用 JAVA 反射机制，而不是字节码增强程序来实现透明性

2.hibernate 性能好，是一个轻量级框架。映射的灵活性很出色。

3.支持各种关系数据库，从一对一到多对多的各种复杂关系。

缺点：限制了使用的对象模型，例如一个持久类不能映射到多个表

3.spring 优缺点

spring 是一个多层的 J2EE 框架，它的目的是提供一种新的机制来管理业务对象以及他们之间的依赖性

优点：1.能够有效的组织你的中间层对象

2.能够消除在许多工程中对 Singleton 的使用

3.通过一种在不同应用程序和项目间一致的方法来处理配置文件，Spring 能消除各种各样自定义格式的属性文件的需要。

4.使用 Spring 构建的应用程序易于单元测试。

5.非侵入性，不需要过多依赖 spring

缺点：配置文件过多

4.说说 Spring 框架的 IOC 和 AOP 概念

Spring 是一种多层的 J2EE 应用程序框架，其核心就是提供一种新的机制管理业务对象及其依赖关系。例如 IOC(控制反转),AOP(面向切面编程)

IOC：Inversion of Control，控制反转。在 Java 开发中，IoC 意味着将你设计好的类交给系统去控制，而不是在你的类内部控制。这称为控制反转。

DI：Dependency Injection 依赖注入

DI 和 IOC 是同一个概念。具体含义是:当某个角色需要另一个角色的协助时，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。但在 Spring 里，创建被调用者的工作不再由调用者来完成，因此称为控制反转;创建被调用者实例的工作通常由 Spring 容器来完成，然后注入调用者，因此也称为依赖注入。

1. Struts 的工作原理

在 Struts 中，用户的请求一般以*.do 作为请求服务名，所有的*.do 请求均被指向 ActionSevlet， ActionSevlet 根据 Struts-config.xml 中的配置信息，将用户请求封装成一个指定名称的 FormBean，并将此 FormBean 传至指定名称的 ActionBean，由 ActionBean 完成相应的业务操作，如文件操作，数据库操作等。每一个*.do 均有对应的 FormBean 名称和 ActionBean 名称，这些在 Struts-config.xml 中配置。

2. Struts 优点与缺点

Struts 是开源软件，使开发者能更深入的了解其内部实现机制。

Struts 优点：业界"标准"（很多成功案例），学习资源丰富。 Struts 的优点主要集中体现在两个方面： Taglib 和页面导航。

a、利用 Struts 提供的 taglib 可以大大节约开发时间。

b、维护扩展比较方便。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处。

c、表现与逻辑分离

d、表单验证解决了请求数据的验证问题，增强了系统健壮性。

e、便于团队开发

Struts 缺点：a、大量的使用标签，对于初学者难度较大。

b、ActionForms 使用不便、无法进行单元测试（StrutsTestCase 只能用于集成）

3. Struts 提供了几个标签库？都是什么标签库？

Struts 提供了五个标签库，即：HTML、Bean、Logic、Template 和 Nested。

HTML 标签 用来创建能够和 Struts 框架和其他相应的 HTML 标签交互的 HTML 输入表单

Bean 标签 在访问 JavaBeans 及其属性，以及定义一个新的 bean 时使用

Logic 标签 管理条件产生的输出和对象集产生的循环

Template 标签 随着 Tiles 框架包的出现，此标记已开始减少使用

Nested 标签 增强对其他的 Struts 标签的嵌套使用的能力

4. Tiles 框架是什么？

Tiles 框架为创建 Web 页面提供了一种模板机制，它能将网页的布局和内容分离。

1、概述 struts，以及 struts 如何实现 MVC 架构的？

答：struts framework 是一种基于 java 的技术，Web 应用程序开发人员通过 struts framework 即可充分利用面向对象设计、代码重用以及"编写一次、到处运行"的优点。Struts 提供了一种创建 Web 应用程序

的框架，其中对应用程序的显示、表示和数据的后端代码进行了抽象。Struts 采用 jsp 作为 MVC 的视图由 ActionServlet 具体指定的 action 动作类作为控制器即 MVC 中的 C，负责视图与模型之间的交互。控制器的每个入口点都由名为 struts-config.xml 的配置文件设置。该文件把来自视图的请求映射为特定的 JAVA 类以进行相应的处理，控制器还指定下一个视图的位置。Struts 中的模型主要指的就是 javabean，它是模型的代表，主要封装数据和业务逻辑。

Struts 的处理流程：

l 控制器进行初始化工作，读取配置文件，为不同的 Struts 模块初始化相应的 ModulConfig 对象。

l 控制器接收 Http 请求，并从 ActionConfig 中找出对应于该请求的 Action 子类，如果没有对应的 Action，控制器直接将请求转发给 JSP 或者静态页面，否则控制器将请求分发至具体的 Action 类进行处理。

l 在控制器调用具体的 Action 的 Execute 方法之前，ActionForm 对象将利用 Http 请求中的参数来填充自己。还可以在 ActionForm 类中调用 Validate 方法来检查请求参数的合法性，并且可以返回一个包含所有错误信息的 Actionerrors 对象。

l 执行具体的的 Execute 的方法，它负责执行相应的业务逻辑。执行完后，返回一个 ActionForward 对象，控制器通过该 ActionForward 对象来进行转发工作。也可以把 Action 要处理的业务逻辑封装在 JavaBean 中，如果系统中还有 EJB，那么通过 JavaBean 调用 EJB 以

完成业务处理；如果没有 EJB，那么就直接在 JavaBean 中连接数据库，进行数据库相关的操作。

2、概述 MVC 体系结构

答：MVC 包括三类对象，model 是应用对象，view 是视图，controller 是控制器，它定义用户界面对用户输入的响应方式。在 MVC 体系中，模型通常被称为"业务逻辑"，是真正完成任务的代码，视图就是使用界面，反映数据的变化。控制器控制着模型和视图之间的交互过程，它决定着向用户返回怎样的视图、检查通过界面输入的信息以及选择处理输入信息的模型

在 MVC 中，表示层和逻辑层分离，各部分可相互独立进行开发，便于开发和维护，提高了开发效率。