

## EJB方面

### 94、EJB2.0有哪些内容?分别用在什么场合? EJB2.0和EJB1.1的区别?

答:规范内容包括Bean提供者, 应用程序装配者, EJB容器, EJB配置工具, EJB服务提供者, 系统管理员。这里面, EJB容器是EJB之所以能够运行的核心。EJB容器管理着EJB的创建, 撤消, 激活, 去活, 与数据库的连接等等重要的核心工作。JSP,Servlet,EJB,JNDI,JDBC,JMS.....

### 95、EJB与JAVA BEAN的区别?

答:Java Bean 是可复用的组件, 对Java

Bean并没有严格的规范, 理论上讲, 任何一个Java类都可以是一个Bean。但通常情况下, 由于Java Bean是被容器所创建(如Tomcat)的, 所以Java Bean应具有一个无参的构造器, 另外, 通常Java Bean还要实现Serializable接口用于实现Bean的持久性。Java

Bean实际上相当于微软COM模型中的本地进程内COM组件, 它是不能被跨进程访问的。Enterprise Java Bean 相当于DCOM, 即分布式组件。它是基于Java的远程方法调用(RMI)技术的, 所以EJB可以被远程访问(跨进程、跨计算机)。但EJB必须被布署在诸如Webspere、WebLogic这样的容器中, EJB客户从不直接访问真正的EJB组件, 而是通过其容器访问。EJB容器是EJB组件的代理, EJB组件由容器所创建和管理。客户通过容器来访问真正的EJB组件。

### 96、EJB是基于哪些技术实现的?并说出SessionBean和EntityBean的区别, StatefulBean和StatelessBean的区别。

答:EJB包括Session Bean、Entity Bean、Message Driven Bean, 基于JNDI、RMI、JAT等技术实现。

SessionBean在J2EE应用程序中被用来完成一些服务器端的业务操作, 例如访问数据库、调用其他EJB组件。EntityBean被用来代表应用系统中用到的数据。

对于客户机, SessionBean是一种非持久性对象, 它实现某些在服务器上运行的业务逻辑。

对于客户机, EntityBean是一种持久性对象, 它代表一个存储在持久性存储器中的实体的对象视图, 或是一个由现有企业应用程序实现的实体。

Session Bean 还可以再细分为 Stateful Session Bean 与 Stateless Session Bean, 这两种的 Session Bean都可以将系统逻辑放在 method之中执行, 不同的是 Stateful Session Bean 可以记录呼叫者的状态, 因此通常来说, 一个使用者会有一个相对应的 Stateful Session Bean 的实体。Stateless Session Bean 虽然也是逻辑组件, 但是他却不负责记录使用者状态, 也就是说当使用者呼叫 Stateless Session Bean 的时候, EJB Container 并不会找寻特定的 Stateless Session Bean 的实体来执行这个 method。换言之, 很可能数个使用者在执行某个 Stateless Session Bean 的 methods 时, 会是同一个 Bean 的 Instance 在执行。从内存方面来看, Stateful Session Bean 与 Stateless Session Bean 比较, Stateful Session Bean 会消耗 J2EE Server 较多的内存, 然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

### 97、EJB与JAVA BEAN的区别?

答:Java Bean 是可复用的组件, 对Java Bean并没有严格的规范, 理论上讲, 任何一个Java类都可以是一个Bean。但通常情况下, 由于Java Bean是被容器所创建(如Tomcat)的, 所以Java Bean应具有一个无参的构造器, 另外, 通常Java Bean还要实现Serializable接口用于实现Bean的持久性。Java Bean实际上相当于微软COM模型中的本地进程内COM组件, 它是不能被跨进程访问的。Enterprise Java Bean 相当于DCOM, 即分布式组件。它是基于Java的远程方法调用(RMI)技术的, 所以EJB可以被远程访问(跨进程、跨计算机)。但EJB必须被布署在诸如Webspere、WebLogic这样的容器中, EJB客户从不直接访问真正的EJB组件, 而是通过其容器访问。EJB容器是EJB组件的代理, EJB组件由容器所创建和管理。客户通过容器来访问真正的EJB组件。

EJB包括(SessionBean,EntityBean)说出他们的生命周期, 及如何管理事务的?

SessionBean: Stateless Session Bean 的生命周期是由容器决定的, 当客户机发出请求要建立一个Bean的实例时, EJB容器不一定要创建一个新的Bean的实例供客户机调用, 而是随便找一个现有的实例提供给客户机。当客户机第一次调用一个Stateful Session Bean 时, 容器必须立即在服务器中创建一个新的Bean实例, 并关联到客户机上, 以后此客户机调用Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的Bean实例。

EntityBean: Entity Beans能存活相对较长的时间, 并且状态是持续的。只要数据库中的数据存在, Entity beans就一直存活。而不是按照应用程序或者服务进程来说的。即使EJB容器崩溃了, Entity beans也是存活的。Entity Beans生命周期能

够被容器或者 Beans自己管理。

EJB通过以下技术管理实务:对象管理组织(OMG)的对象实务服务(OTS), Sun Microsystems的Transaction Service(JTS)、Java Transaction API(JTA), 开发组(X/Open)的XA接口。

#### 98、EJB的角色和三个对象

答:一个完整的基于EJB的分布式计算结构由六个角色组成, 这六个角色可以由不同的开发商提供, 每个角色所作的工作必须遵循Sun公司提供的EJB规范, 以保证彼此之间的兼容性。这六个角色分别是EJB组件开发者(Enterprise Bean Provider)、应用组合者(Application Assembler)、部署者(Deployer)、EJB 服务器提供者(EJB Server Provider)、EJB 容器提供者(EJB Container Provider)、系统管理员(System Administrator)

三个对象是Remote(Local)接口、Home(LocalHome)接口, Bean类

#### 99、EJB容器提供的服务

答:主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发行管理等服务。

#### 100、EJB规范规定EJB中禁止的操作有哪些?

答:1.不能操作线程和线程API(线程API指非线程对象的方法如notify,wait等), 2.不能操作awt, 3.不能实现服务器功能, 4.不能对静态属生存取, 5.不能使用IO操作直接存取文件系统, 6.不能加载本地库., 7.不能将this作为变量和返回, 8.不能循环调用。

#### 101、remote接口和home接口主要作用

答:remote接口定义了业务方法, 用于EJB客户端调用业务方法。

home接口是EJB工厂用于创建和移除查找EJB实例

#### 102、bean 实例的生命周期

答:对于Stateless Session Bean、Entity Bean、Message Driven Bean一般存在缓冲池管理, 而对于Entity Bean和Stateful Session Bean存在Cache管理, 通常包含创建实例, 设置上下文、创建EJB Object(create)、业务方法调用、remove等过程, 对于存在缓冲池管理的Bean, 在create之后实例并不从内存清除, 而是采用缓冲池调度机制不断重用实例, 而对于存在Cache管理的Bean则通过激活和去激活机制保持Bean的状态并限制内存中实例数量。

#### 103、EJB的激活机制

答:以Stateful Session Bean 为例:其Cache大小决定了内存中可以同时存在的Bean实例的数量, 根据MRU或NRU算法, 实例在激活和去激活状态之间迁移, 激活机制是当客户端调用某个EJB实例业务方法时, 如果对应EJB Object发现自己没有绑定对应的Bean实例则从其去激活Bean存储中(通过序列化机制存储实例)回复(激活)此实例。状态变迁前会调用对应的ejbActive和ejbPassivate方法。

#### 104、EJB的几种类型

答:会话(Session)Bean, 实体(Entity)Bean 消息驱动的(Message Driven)Bean

会话Bean又可分为有状态(Stateful)和无状态(Stateless)两种

实体Bean可分为Bean管理的持续性(BMP)和容器管理的持续性(CMP)两种

#### 105、客户端调用EJB对象的几个基本步骤

答:设置JNDI服务工厂以及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接口调用其业务方法。

## 应用服务器方面

#### 106、如何给weblogic指定大小的内存?

答:在启动Weblogic的脚本中(位于所在Domian对应服务器目录下的startServerName), 增加set MEM\_ARGS=-Xms32m -Xmx200m, 可以调整最小内存为32M, 最大200M

EJB需直接实现它的业务接口或Home接口吗, 请简述理由。

远程接口和Home接口不需要直接实现, 他们的实现代码是由服务器产生的, 程序运行中对应实现类会作为对应接口类型的实例被使用。

#### 107、应用服务器有那些?

答:BEA WebLogic Server, IBM WebSphere Application Server, Oracle9i Application Server, jBoss, Tomcat

#### 108、如何设定的weblogic的热启动模式(开发模式)与产品发布模式?

答:可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者commenv文件, 增加set PRODUCTION\_MODE=true。

#### 109、如何启动时不需输入用户名与密码?

答:修改服务启动文件, 增加 WLS\_USER和WLS\_PW项。也可以在boot.properties文件中增加加密过的用户名和密码。

#### 110、在weblogic管理控制台中对一个应用域(或者说是一个网站,Domain)进行jms及ejb或连接池等相关信息进行配置后,实际保存在什么文件中?

答:保存在此Domain的config.xml文件中, 它是服务器的核心配置文件。

#### 111、说说weblogic中一个Domain的缺省目录结构?比如要将一个简单的helloWorld.jsp放入何目录下,然的在浏览器上就可打入 aspectratio="t" v:ext="edit"><http://主机:端口号/helloworld.jsp>就可以看到运行结果了?

又比如这其中用到了一个自己写的javaBean该如何办?

答:Domain目录服务器目录applications, 将应用目录放在此目录下将可以作为应用访问, 如果是Web应用, 应用目录需要满足Web应用目录要求, jsp文件可以直接放在应用目录中, Javabeen需要放在应用目录的WEB-INF目录的classes目录中, 设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

#### 112、在weblogic中发布ejb需涉及到哪些配置文件

答:不同类型的EJB涉及的配置文件不同, 都涉及到的配置文件包括ejb-jar.xml,weblogic-ejb-jar.xmlCMP实体Bean一般还需要weblogic-cmp-rdbms-jar.xml

#### 86、如何在weblogic中进行ssl配置与客户端的认证配置或说说j2ee(标准)进行ssl的配置

缺省安装中使用Demoidentity.jks和DemoTrust.jks KeyStore实现SSL, 需要配置服务器使用Enable SSL, 配置其端口, 在产品模式下需要从CA获取私有密钥和数字证书, 创建identity和trust keystore, 装载获得的密钥和数字证书。可以配置此SSL连接是单向还是双向的。

#### 113、如何查看在weblogic中已经发布的EJB?

答:可以使用管理控制台, 在它的Deployment中可以查看所有已发布的EJB

说说在weblogic中开发消息Bean时的persistent与非-persistent的差别

persistent方式的MDB可以保证消息传递的可靠性,也就是如果EJB容器出现问题而JMS服务器依然会将消息在此MDB可用的时候发送过来, 而非-persistent方式的消息将被丢弃。

## J2EE,MVC方面

#### 114、MVC的各个部分都有那些技术来实现?如何实现?

答:MVC是Model—View—Controller的简写。"Model" 代表的是应用的业务逻辑(通过JavaBean, EJB组件实现), "View" 是应用的表示面(由JSP页面产生), "Controller"

是提供应用的处理过程控制(一般是一个Servlet), 通过这种设计模型把应用逻辑, 处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

#### 115、J2EE是什么?

答:Je22是Sun公司提出的多层(multi-tiered),分布式(distributed),基于组件(component-base)的企业级应用模型(enterprise application

model)。在这样的一个应用系统中, 可按照功能划分为不同的组件, 这些组件又可在不同计算机上, 并且处于相应的层次(tier)中。所属层次包括客户层(client tier)组件,web层和组件,Business层和组件,企业信息系统(EIS)层。

#### 116、WEB SERVICE名词解释。JSWDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI,WSDL解释。

答:Web Service描述语言WSDL

SOAP即简单对象访问协议(Simple Object Access Protocol), 它是用于交换XML编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准;UDDI是一套基于Web的、分布式的、为Web

Service提供的、信息注册中心的实现标准规范, 同时也包含一组使企业能将自身提供的Web

Service注册, 以使别的企业能够发现的访问协议的实现标准。

#### 117、J2EE是技术还是平台还是框架？

答: J2EE本身是一个标准, 一个为企业分布式应用的开发提供的标准平台。

J2EE也是一个框架, 包括JDBC、JNDI、RMI、JMS、EJB、JTA等技术。

#### 118、STRUTS的应用(如STRUTS架构)

答: Struts是采用Java Servlet/JavaServer Pages技术, 开发Web应用程序的开放源码的framework。

采用Struts能开发出基于MVC(Model-View-Controller)设计模式的应用构架。Struts有如下的主要功能:

- 一. 包含一个controller servlet, 能将用户的请求发送到相应的Action对象。
- 二. JSP自由tag库, 并且在controller servlet中提供关联支持, 帮助开发员创建交互式表单应用。
- 三. 提供了一系列实用对象: XML处理、通过Java reflection APIs自动处理JavaBeans属性、国际化的提示和消息。

#### 119、WEB SERVICE名词解释。JSDDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI,WSDL解释。

答: Web Service Web Service是基于网络的、分布式的模块化组件, 它执行特定的任务, 遵守具体的技术规范, 这些规范使得Web Service能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了Java中使用DOM, SAX, XSLT的通用的接口。这样在你的程序中你只要使用这些通用的接口, 当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging) 是为SOAP通信提供访问方法和传输机制的API。

WSDL是一种 XML 格式, 用于将网络服务描述为一组端点, 这些端点对包含面向文档信息或面向过程信息的信息进行操作。这种格式首先对操作和消息进行抽象描述, 然后将其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为抽象端点(服务)。

SOAP即简单对象访问协议(Simple Object Access Protocol), 它是用于交换XML编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准; UDDI是一套基于Web的、分布式的、为Web Service提供的、信息注册中心的实现标准规范, 同时也包含一组使企业能将自身提供的Web Service注册, 以使别的企业能够发现的访问协议的实现标准。

#### 120、C/S 与 B/S 区别:

答: 有如下八个方面的不同:

##### 1. 硬件环境不同:

☐ C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。

☐

B/S 建立在广域网之上的, 不必是专门的网络硬件环境, 例与电话上网, 租用设备. 信息自己管理. 有比C/S更强的适应范围, 一般只要有操作系统和浏览器就行

##### 2. 对安全要求不同

☐

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用C/S 结构适宜. 可以通过B/S发布部分可公开信息.

☐ B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

##### 3. 对程序架构不同

☐ C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑。

☐

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比C/S有更高的要求 B/S结构的程序架构是发展的趋势, 从MS的.Net系列的BizTalk 2000 Exchange 2000等, 全面支持网络的构件搭建的系统. SUN 和IBM推的JavaBean 构件技术等, 使 B/S更加成熟。

##### 4. 软件重用不同

☐ C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在B/S要求下的构件的重用性好。

☐

B/S 对的多重结构, 要求构件相对独立的功能. 能够相对较好的重用. 就入买来的餐桌可以再利用, 而不是做在墙上的石头桌子

##### 5. 系统维护不同



□□C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

□□

B/S 构件组成, 方面构件个别的更换, 实现系统的无缝升级. 系统维护开销减到最小. 用户从网上自己下载安装就可以实现升级.

#### 6. 处理问题不同

□□C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

□□B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S无法作到的. 与操作系统平台关系最小.

#### 7. 用户接口不同

□□C/S 多是建立的Window平台上, 表现方法有限, 对程序员普遍要求较高

□□B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低, 减低开发成本.

#### 8. 信息流不同

□□C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

□□B/S 信息流向可变化, B-B B-C B-G等信息、流向的变化, 更像交易中心。

### 121、什么是web容器

答: 给处于其中的应用程序组件(JSP, SERVLET)提供一个环境, 使JSP, SERVLET直接更容器中的环境变量接\*\*互, 不必关注其它系统问题. 主要有WEB服务器来实现. 例如: TOMCAT, WEBLOGIC, WEBSPPHERE等. 该容器提供的接口严格遵守J2EE规范中的WEB APPLICATION 标准. 我们把遵守以上标准的WEB服务器就叫做J2EE中的WEB容器.

### 122、什么是EJB容器

答: Enterprise java bean 容器. 更具有行业领域特色. 他提供给运行在其中的组件EJB各种管理功能. 只要满足J2EE规范的EJB放入该容器, 马上就会被容器进行高效率的管理. 并且可以通过现成的接口来获得系统级别的服务. 例如邮件服务、事务管理

### 123、什么是JNDI

答: (Java Naming & Directory Interface) JAVA命名目录服务. 主要提供的功能是: 提供一个目录系统, 让其它各地的应用程序在其上面留下自己的索引, 从而满足快速查找和定位分布式应用程序的功能。

### 124、什么是JMS

答: (Java Message Service) JAVA消息服务. 主要实现各个应用程序之间的通讯. 包括点对点和广播。

### 125、什么是JTA

答: (Java Transaction API) JAVA事务服务. 提供各种分布式事务服务. 应用程序只需调用其提供的接口即可。

### 126、什么是JAF

答: (Java Action FrameWork) JAVA安全认证框架. 提供一些安全控制方面的框架. 让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP: (Remote Method Invocation /internet对象请求中介协议) 他们主要用于通过远程调用服务. 例如, 远程有一台计算机上运行一个程序, 它提供股票分析服务, 我们可以在本地计算机上实现对其直接调用. 当然这是要通过一定的规范才能在异构的系统之间进行通信. RMI是JAVA特有的。

### 127、MVC的各个部分都有那些技术来实现? 如何实现?

答: MVC是Model—View—Controller的简写. "Model" 代表的是应用的业务逻辑(通过JavaBean, EJB组件实现), "View" 是应用的表示面(由JSP页面产生), "Controller" 是提供应用的处理过程控制(一般是一个Servlet), 通过这种设计模型把应用逻辑, 处理过程和显示逻辑分成不同的组件实现. 这些组件可以进行交互和重用。

## 设计模式方面

### 128、开发中都用到了那些设计模式? 用在什么场合?

答: 每个模式都描述了一个在我们的环境中不断出现的问题, 然后描述了该问题的解决方案的核心. 通过这种方式, 你可以无数次地使用那些已有的解决方案, 无需在重复相同的工作. 主要用到了MVC的设计模式. 用来开发JSP/Servlet或者J2EE的相关应用. 简单工厂模式等。

**129、说说你所熟悉或听说过的j2ee中的几种常用模式?及对设计模式的一些看法**

答:Session Facade Pattern:使用SessionBean访问EntityBean

Message Facade Pattern:实现异步调用

EJB Command Pattern:使用Command JavaBeans取代SessionBean, 实现轻量级访问

Data Transfer Object Factory:通过DTO Factory简化EntityBean数据提供特性

Generic Attribute Access:通过AttributeAccess接口简化EntityBean数据提供特性

Business Interface:通过远程(本地)接口和Bean类实现相同接口规范业务逻辑一致性

EJB架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂, 项目队伍越庞大则越能体现良好设计的重要性。

**130、j2ee常用的设计模式?说明工厂模式。**

答:Java中的23种设计模式:

Factory(工厂模式), Builder(建造模式), Factory Method(工厂方法模式),

Prototype(原始模型模式), Singleton(单例模式), Facade(门面模式),

Adapter(适配器模式), Bridge(桥梁模式), Composite(合成模式),

Decorator(装饰模式), Flyweight(享元模式), Proxy(代理模式),

Command(命令模式), Interpreter(解释器模式), Visitor(访问者模式),

Iterator(迭代子模式), Mediator(调停者模式), Memento(备忘录模式),

Observer(观察者模式), State(状态模式), Strategy(策略模式),

Template Method(模板方法模式), Chain Of Responsibility(责任链模式)

工厂模式:工厂模式是一种经常被使用到的模式, 根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例, 通常这一组类有一个公共的抽象父类并且实现了相同的方法, 但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类, 该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类, 工厂类可以根据条件生成不同的子类实例。当得到子类的实例后, 开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

**131、UML方面**

答:标准建模语言UML。用例图,静态图(包括类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图,

UML方面

## CORBA方面

**132、CORBA是什么?用途是什么?**

答:CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture), 由对象管理组织(Object Management Group, 缩写为 OMG)标准化。它的组成是接口定义语言(IDL),

语言绑定(binding:也译为联编)和允许应用程序间互操作的协议。其目的为:

用不同的程序设计语言书写

在不同的进程中运行

为不同的操作系统开发

## LINUX方面

**133、LINUX下线程, GDI类的解释。**

答:LINUX实现的就是基于核心轻量级进程的“一对一”线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI类为图像设备编程接口类库。

**134、LINUX下线程, GDI类的解释。**

LINUX实现的就是基于核心轻量级进程的“一对一”线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理

---

在核外函数库中实现。

GDI类为图像设备编程接口类库。

动力节点