教师用习题书

# 第一部分：Web核心协议与架构

1. 请给出作为一种互联网应用，Web的核心协议和采用的技术是什么？
 1）URI标识资源

2）基于B/S模式

3）使用HTML技术来创建客户端界面

4）使用HTTP协议来传输内容

1. 请给出URI和URL的概念，以及区别和联系。
 URI:统一资源标识符（Uniform Resource Identifier），URL:统一资源定位符（Uniform Resource Locator ）。URI比较偏高层次和抽象，用于唯一确定互联网上的资源。URL是URI的一种实现，以名称的方式唯一确定资源的URN是URI的另外一种实现。URL表现形式上是用来确定Web网上某资源地址的字符串。
2. 使用一个典型的浏览器，并通过[**http://html5test.com/**](http://html5test.com/)网站获取你设备（包括移动设备）和浏览器对html5标准的支持程度，进行总结比较。并初步了解html5标准的各种组成部分和功能。

 同一个设备上，不同浏览器的实现对html5的实现支持也是不同的。比如目前移动设备上支持较好的为Opera浏览器。

1. 使用一个典型的浏览器，并通过互联网搜索引擎查找信息，获取一种工具或者浏览器功能，能检测到访问一个网站时，运用http协议的请求和响应信息。

 在Chrome浏览器上按F12，然后选择“网络”可以捕获到访问一个URL的时候的http请求和响应信息。

1. 使用支持http2, https, QUIC, SPDY等协议的浏览器和工具，进行实践。并对比这些协议对传统http的改进和探索，并了解这些协议最新的发展。

 在Chrome浏览器上



1. 描述基于web的应用程序的特征。

（1）以浏览器作为用户界面的窗口

（2）以可交互的网页形式提供用户界面，网页由HTML语言写成

（3）提供与普通application类似的功能

（4）使用browser-server架构

（5）browser与server之间通讯基于http协议

1. 描述一个完整的web应用程序，从前端到后端处理的流程。



1. 给出从文档web，到应用web，到服务web的各自特征，以及核心协议。

2. 通过互联网搜索资料和自主学习，你认为web进一步发展的方向和趋势是什么？

Semantic web, web VR, 物联网与Web技术的结合，云化

1. 比较C/S模式(胖客户端)和B/S模式（瘦客户端）的网络程序的主要特征和异同，以实例进行说明。
* C/S: **优点**
	+ 提供给终端用户更多的功能，使得终端用户根据自己的需要配置应用程序，因为胖客户机可以存储客户系统的大部分个人数据；
	+ 可以减少服务器的负担，因为复杂的计算操作是由客户端自己完成

 **缺陷**

* + 需要更多的可能导致错误的安装过程
	+ 使用起来需要教育培训
	+ 对新版本的软件需要重新安装
	+ 需要较多的客户端资源（如内存和CPU处理能力）

B/S

 **优点**

* + 易于部署
	+ 易于使用
	+ 通过集中管理使系统管理更容易
		- **通过集中管理和监督可以很容易地发现问题**
		- **在服务器端进行问题的解决**
		- **新版本的软件只需安装在服务器上**
	+ 瘦客户使用的客户端资源很少

 **缺点**

* + 实时性较差
	+ 界面用户体验较差
1. Web本质上是一种分布式计算，简要阐述分布式计算的主要发展阶段。

主机系统 -> C/S ->B/S ->P2P

# 第二部分：Web2.0概念与核心技术

1. 请阐述以下概念和技术的含义和特征：（A）RIA；（B）Web2.0；（C）中间件；（D）SOA；（E）N层架构

A、RIA

RIA:富客户端网络应用（Rich Web Applications）, 结合C/S和B/S的优点，具有以下特征：

数据模型的丰富：用户界面可以显示和操作更为复杂的嵌入在客户端的数据模型

用户界面的丰富：提供了灵活多样的界面控制元素，这些控制元素可以很好的与数据模型相结合

最好的通讯模式：无刷新页面之下提供快捷的界面响应时间，双向互动声音和图像。

B、Web2.0

Web2.0是以Blog、TAG、SNS、RSS、wiki等应用为核心，依据六度分隔、xml、ajax等新理论和技术实现的互联网新一代模式

C、中间件

中间件是是操作系统和应用软件之间的一个独立软件层。它提供服务用来协调不同的应用软件。

中间件通过提供简单、一致、集成的分布编程环境，简化分布应用的设计、编程和管理。本质上，中间件是一个分布软件层，抽象了底层分布环境（网络、主机、操作系统、编程语言）的复杂性和异构性。

D、SOA

面向服务的架构是一种设计模式，它指导着业务服务（Business Services）在其生命周期（从构思开始，直至停止使用）中包括创建和使用的方方面面。SOA也是一种定义和提供IT基础设计（IT Infrastructure）的方式，它允许不同应用相互交换数据、参与业务流程（Business Processes），无论它们各自背后使用的是何种操作系统或采用了何种编程语言

E、N层架构

在3层结构中，客户层和数据层已被严格定义，但中间层并未明确定义。

中间层可以包括所有与应用程序的界面和持久数据存储无关的处理。

假定将中间层划分成许多服务程序是符合逻辑的，那么将每一主要服务都视为独立的层，则3层结构就成为了n层结构。

1. XML，HTML，SGML的区别和联系以及各自特点

XML和HTML都源自SGML， HTML是SGML的一个应用，XML是SGML的一个子集。

ＳＧＭＬ是标准通用化标记语言（Standard Generalized Markup Language）的简称，是一种用标记（tag）来描述文档资料的国际标准通用语言。

HTML用于浏览器解析并显示信息，主要给人观看。而XML用于以文档为中心的应用和以数据为中心的应用，其特性有：

纯文本

易创建和编辑

提供了从小型配置文件到数据库XML前端的良好可伸缩性

数据标识给予了语义信息基础

内容与表示分开，同样的内容可以不同的表示

结构化，使得XML文档数据具有模块化特征

丰富的链接功能

独立于平台和语言

容易处理，来自于严格的语法和标签机制

1. 回答关于XML的相关问题：

（1）相对于HTML的不足而言，XML的优势在什么地方？

（2）简述XML的规范DTD, Schema，以及XSLT分别起到什么作用

（3）Java处理XML主要有DOM和SAX，简述其特征

（1）

可扩展性

对文档内容具有自描述能力，支持智能代码和智能搜索

结构化数据表达能力

良好的通用数据格式，跨平台以及语言独立

强大的超链接功能

文档的表示形式多样化

降低了对服务端的要求

（2）DTD和Schema分别规范了XML文档里面可以出现的节点类型，以及相互的关系，XSLT给出了XML数据变换的规范

（3）SAX 这种基于事件的 API 中，解析器发送一些事件给某类侦听器，对内存要求不高，解析速度快，但是不能随机访问。在如 DOM 这种基于树的 API 中，解析器在内存中构建数据树，对内存要求比较高，需要全部装载到内存中，可以实现随机访问。

1. 简单的描述一下组件技术的模型，并且列举几个当前具有代表性的技术，在此基础上解释一下Java Bean的定义、特性和优点

组件是自包含的、可重用的软件单元

软件组件模型描述了如何开发和使用组件。软件组件模型包括以下元素：

1 创造和使用组件的规范；

2 软件实现规范，为建造和使用组件提供一种具体的机制。

目前具备代表性的组件技术有

微软的COM、COM+，

Sun的JavaBean和EJB（Enterprise Java Bean） JavaBean规范为Java开发人员提供了一种“组件化”其Java类的方法

CORBA（Common Object Request Broker Architecture，公共对象请求代理结构）

在Sun公司的Java Bean规范的定义中，Bean的定义是：“Bean是一个基于Sun公司的JavaBean规范的、可在编程工具中被可视化处理的可复用的软件组件”。

JavaBean是基于Sun公司的JavaBean规范的，可在编程工具中被可视化处理的可复用的软件组件。因此JavaBean具有4个基本特性：

独立性

可重用性

在可视化开发工具中使用

状态可以保存

JavaBean分成可视组件和非可视组件。在JSP中主要使用非可视组件。 便于维护代码；降低开发JSP网页人员对Java编程能力的要求。

1. 以Project中验证用户名有效性为例简述一下AJAX工作流程（步骤）。

1.A client event occurs

2.An XMLHttpRequest object is created

3.The XMLHttpRequest object is configured

4.The XMLHttpRequest object makes an async. request

5.The ValidateServlet returns an XML document containing the result

6.The XMLHttpRequest object calls the callback() function and processes the result

7.The HTML DOM is updated

1. 请说明Ajax的基本原理，以及基于ajax的web应用程序与传统的web应用程序的区别

Ajax是Asynchronous JavaScript and XML的缩写

有确定需要从服务器读取新数据时再由Ajax引擎代为向服务器提交请求。

使用XHTML和CSS标准化呈现

使用DOM实现动态显示和交互

使用XML和XSLT进行数据交换与处理

使用XMLHttpRequest进行异步数据读取，向XMLHttpRequest注册一个回调函数

最后用JavaScript整合以上技术

 区别： Ajax可以实现异步调用，局部更新，增加了用户体验。

7、 简要介绍一下Web2.0的相关应用：（1）Blog；（2）RSS；（3）WIKI；（4）SNS；（5）P2P。

（1）Blog——博客/网志

　　Blog的全名应该是Web log，后来缩写为Blog。Blog是一个易于使用的网站，您可以在其中迅速发布想法、与他人交流以及从事其他活动。

RSS

（2）RSS

RSS是站点用来和其他站点之间共享内容的一种简易方式（也叫聚合内容）的技术。最初源自浏览器“频道”的技术，现在通常被用于新闻和其他按顺序排列的网站，例如Blog。

（3）Wiki

　　Wiki——一种多人协作的写作工具。Wiki站点可以有多人（甚至任何访问者）维护，每个人都可以发表自己的意见，或者对共同的主题进行扩展或者探讨。

　　Wiki指一种超文本系统。这种超文本系统支持面向社群的协作式写作，同时也包括一组支持这种写作的辅助工具。有人认为，Wiki系统属于一种人类知识网格系统，我们可以在Web的基础上对Wiki文本进行浏览、创建、更改，而且创建、更改、发布的代价远比HTML文本小；同时Wiki系统还支持面向社群的协作式写作，为协作式写作提供必要帮助；最后，Wiki的写作者自然构成了一个社群，Wiki系统为这个社群提供简单的交流工具。与其它超文本系统相比，Wiki有使用方便及开放的特点，所以Wiki系统可以帮助我们在一个社群内共享某领域的知识。

（4）SNS

　　SNS，Social Network Sofwaret，社会性网络软件，依据六度理论，以认识朋友的朋友为基础，扩展自己的人脉。

（5）P2P

　　P2P是peer-to-peer的缩写，peer在英语里有“（地位、能力等）同等者”、“同事”和“伙伴”等意义。这样一来，P2P也就可以理解为“伙伴对伙伴”、“点对点”的意思，或称为对等联网。目前人们认为其在加强网络上人的交流、文件交换、分布计算等方面大有前途。

1. Web1.0与Web2.0之间的对比比较？

Web2.0 是相对Web1.0 的新的一类互联网应用的统称。Web1.0 的主要特点在于用户通过浏览器获取信息。Web2.0 则更注重用户的交互作用，用户既是网站内容的浏览者，也是网站内容的制造者。所谓网站内容的制造者是说互联网上的每一个用户不再仅仅是互联网的读者，同时也成为互联网的作者；不再仅仅是在互联网上冲浪，同时也成为波浪制造者;在模式上由单纯的“读”向“写”以及“共同建设”发展；由被动地接收互联网信息向主动创造互联网信息发展，从而更加人性化！

1. 简要介绍Web 2.0的概念？

Web2.0是以Blog、TAG、SNS、RSS、Wiki等应用为核心，依据六度分隔、XML、Ajax等新理论和技术实现的互联网新一代模式。

# 第三部分 前端框架angular

1. 请阐述流行的前端框架，以及它们的主要特点。

Google的Angular，Facebook的React，个人维护的Vue。

后面两者主要是虚拟dom，实现视图与模型数据的绑定。而Google的Angular是一个较为全面的前端框架。Vue较为容易学习，适合在其基础上加入其它客户定制组件; React使用了JSX语法，学习曲线较难。相比于Vue,Angular学习曲线较难。

1. 列举两个与Angular相关的Google前端技术。

Material: 一种最初用于Android上的界面风格设计，后来实现了Android,平板，PC等跨平台设计的扁平式设计的UI框架。

Ionic: 一个与Angular兼容的移动前端框架。

1. Angular与AngularJS相比的特点和优势在什么地方？

更好的速度和性能

更简单的依赖注入

模块化

具备ES6和Typescript的优点。

灵活的路由，具备延迟载入功能

更易于学习

1. 阐述Angular 2中的路由工作原理。
 路由是能够让用户在视图/组件之间导航的机制。Angular 2简化了路由，并提供了在模块级（延迟载入）下配置和定义的灵活性。

Angular应用程序具有路由器服务的单个实例。而且每当URL改变时。对应的路由就与路由配置数组进行匹配。在成功匹配时，它会应用重定向，此时路由器会构建ActivatedRoute对象的树。同一时候包括路由器的当前状态。在重定向之前，路由器将通过执行保护（CanActivate）来检查是否同意新的状态。

路由保护仅仅是路由器执行来检查路由授权的接口方法。

执行后，它将解析路由数据并通过将所需的组件实例化到<router-outlet> </ router-outlet>中来激活路由器状态。

1. 阐述Angular 2中的延迟载入

延迟载入使我们仅仅载入用户正在交互的模块，而其余的模块会在执行时按需载入。延迟载入通过将代码拆分成多个包并以按需载入的方式，来加速应用程序初始载入过程。每一个Angular应用程序必须有一个叫AppModule的主模块。代码应该依据应用程序业务案例分为不同的子模块（NgModule）。

1. 请解释Angular 2应用程序的生命周期hooks是什么？

Angular 2组件/指令具有生命周期事件。是由@angular/core管理的。

@angular/core会创建组件。渲染它。创建并呈现它的后代。当@angular/core的数据绑定属性更改时，处理就会更改，在从DOM中删除其模板之前，就会销毁掉它。Angular提供了一组生命周期hooks（特殊事件）。能够被分接到生命周期中，并在须要时执行操作。

构造函数会在全部生命周期事件之前执行。每一个接口都有一个前缀为ng的hook方法。比如，ngOnint界面的OnInit方法，这种方法必须在组件中实现。 一部分事件适用于组件/指令，而少数事件仅仅适用于组件。

ngOnChanges：当Angular设置其接收当前和上一个对象值的数据绑定属性时响应。

ngOnInit：在第一个ngOnChange触发器之后，初始化组件/指令。这是最经常使用的方法。用于从后端服务检索模板的数据。

ngDoCheck：检測并在Angular上下文发生变化时执行。每次更改检測执行时，会被调用。

ngOnDestroy：在Angular销毁指令/组件之前清除。取消订阅可观察的对象并脱离事件处理程序，以避免内存泄漏。

组件特定hooks：

ngAfterContentInit：组件内容已初始化完毕。

ngAfterContentChecked：在Angular检查投影到其视图中的绑定的外部内容之后。

ngAfterViewInit：Angular创建组件的视图后。

ngAfterViewChecked：在Angular检查组件视图的绑定之后。

1. 在Angular 2应用中，我们应该注意哪些安全威胁？

就像不论什么其他client或Web应用程序一样，Angular 2应用程序也该遵循一些基本准则来减轻安全风险。

当中一些是：

1.避免为你的组件使用/注入动态HTML内容‘；

2.假设使用外部HTML，也就是来自数据库或应用程序之外的地方。那么须要清理它；

3.不要将外部网址放在应用程序中，除非它是受信任的。避免网址重定向。除非它是可信的；

4.考虑使用AOT编译或离线编译；

5.通过限制api。选择使用已知或安全环境/浏览器的app来防止XSRF攻击。

1. 怎样优化Angular 2应用程序来获得更好的性能？

优化取决于应用程序的类型和大小以及更多因素。

但一般来说，在优化Angular 2应用程序时。会考虑下面几点：

1.考虑AOT编译；

2.确保应用程序已经经过了捆绑。uglify和tree shaking；

3.确保应用程序不存在不必要的import语句；

4.确保应用中已经移除了不使用的第三方库；

5.全部dependencies 和dev-dependencies都是明白分离的；

6.假设应用程序较大时，我会考虑延迟载入而不是全然捆绑的应用程序。

1. 什么是AOT编译？它有什么优缺点？

AOT编译代表的是Ahead Of Time编译，当中Angular编译器在构建时，会将Angular组件和模板编译为本机JavaScript和HTML。编译好的HTML和JavaScript将会部署到Webserver，以便浏览器能够节省编译和渲染时间。

优点：

1.更快的下载：由于应用程序已经编译。很多Angular编译器相关库就不再须要捆绑，应用程序包变得更小，所以该应用程序能够更快地下载。

2.更少的Http请求数：假设应用程序没有捆绑来支持延迟载入（或不论什么原因），对于每一个关联的HTML和CSS，都会有一个单独的server请求。可是预编译的应用程序会将全部模板和样式与组件对齐，因此到server的Http请求数量会更少。

3.更快的渲染：假设应用程序不是AOT编译，那么应用程序全然载入时，编译过程会发生在浏览器中。这须要等待下载全部必需的组件。然后等待编译器花费时间来编译应用程序。使用AOT编译，就能实现优化。

4.在构建时检測错误：由于预先编译，能够检測到很多编译时错误，能够为应用程序提供更好的稳定性。

缺点：

1.仅适用于HTML和CSS，其他文件类型须要前面的构建步骤

2.没有watch模式。必须手动完毕（bin / ngc-watch.js）并编译全部文件

3.须要维护AOT版本号的bootstrap文件（使用cli等工具时不须要）

在编译之前，须要清理步骤

1. Observables和Promises的核心差别是什么？

从堆栈溢出就是一个差别：

当异步操作完毕或失败时。Promise会处理一个单个事件。

Observable相似于（在很多语言中的）Stream，当每一个事件调用回调函数时。同意传递零个或多个事件。通常Observable比Promise更受欢迎。由于它不但提供了Promise特性，还提供了其他特性。

使用Observable能够处理0,1或多个事件。你能够在每种情况下使用同样的API。Observable是可取消的，这相比于Promise也具有优势。

假设server的HTTP请求结果或其他一些异步操作不再须要，则Observable的订阅者能够取消订阅，而Promise将终于调用成功或失败的回调，即使你不须要通知或其提供的结果。

Observable提供像map。forEach，reduce之类的相似于数组的运算符，还有强大的运算符，如retry（）或replay（）等，使用起来是相当方便的。

Promises vs Observables

Promises：

返回单个值

不可取消

Observables：

能够使用多个值

可取消

支持map，filter，reduce和相似的操作符

ES 2016提议的功能

使用反应式扩展（RxJS）

依据时间的变化，数组成员能够异步获取

1. 什么是Shadow DOM？它怎样帮助Angular 2更好地执行？

Shadow DOM是HTML规范的一部分。它同意开发者封装自己的HTML标记，CSS样式和JavaScript。

Shadow DOM以及其他一些技术，使开发者能够像<audio>标签一样构建自己的一级标签。Web组件和API。总的来说，这些新的标签和API被称为Web组件。Shadow DOM通过提供了更好的关注分离，通过其他的HTML DOM元素实现了更少的样式与脚本的冲突。

由于Shadow DOM本质上是静态的。同一时候也是开发者无法訪问的，所以它是一个非常好的候选对象。由于它缓存的DOM将在浏览器中呈现得更快，并提供更好的性能。此外。还能够相对非常好地管理shadow DOM。同一时候检測Angular 2应用的改变，而且能够有效地管理视图的又一次绘制。

# 第四部分 后端框架

1. EJB是基于哪些技术实现的？并请说明Session Bean和Entity Bean的区别，Stateful Bean和Stateless Bean的区别。

EJB是基于分布式对象技术、服务端构件技术、以及CTM技术实现。SessionBean一般代表一个操作，不需要持久化； Entity Bean代表一个实体，一般需要持久化。Stateful Bean具有状态维护信息，对于不同用户需要不同的Bean， 而stateless bean没有维护状态信息，可以对应各个用户。

1. MVC模式的含义是什么（可画简图示意），其优点是什么？

模型-视图-控制）模式，模型层负责表达和访问商业数据，执行商业逻辑和操作；视图层负责显示模型层的内容,视图层也会将用户的输入传送给控制器。控制层负责定义应用程序的行为。



优点：在模型层、视图层和控制层之间划分责任可以减少代码的重复度，并使应用程序维护起来更简单。

由于数据和商务逻辑的分开，在新的数据源加入和数据显示变化的时候，数据处理也会变得更简单 。

实现了功能模块和显示模块的分离，同时它还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性

1. 在使用JSP，Servlet和javabean技术的情况下，他们分别充当MVC的什么角色。并请画出浏览器端提出申请，到服务器端处理该申请并返回结构的示意图。

MVC的优点

在模型层、视图层和控制层之间划分责任可以减少代码的重复度，并使应用程序维护起来更简单 ；

由于数据和商务逻辑的分开，在新的数据源加入和数据显示变化的时候，数据处理也会变得更简单 ；

实现了功能模块和显示模块的分离，同时它还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性；

JSP充当视图层，Servlet充当控制层，javabean充当模型层。



1. 请叙述JSP MODEL1和 JSP MODEL2两种框架的含义，以及各自的特点。

Model1，JSP+JavaBean设计模式，是以页面为中心的架构：在这种模式中，JSP页面独自响应请求并将处理结果返回客户，所有的数据库操作通过JavaBean来实现。

大量地使用这种模式，常会导致在JSP页面中嵌入大量的Java代码，当需要处理的商业逻辑非常复杂时，这种情况就会变得很糟糕。大量的Java代码使得JSP页面变得非常臃肿。前端的页面设计人员稍有不慎，就有可能破坏关系到商业逻辑的代码。这种情况在大型项目中经常出现，造成了代码开发和维护的困难，同时会导致项目管理的困难。因此这种模式只适用于中小规模的项目。



Model2，MVC设计模式，以JSP技术实现视图的功能，用Servlet技术实现控制器的功能，用JavaBean技术或者EJB实现模型的功能。



1. Java操作XML的处理器主要有哪两种？他们的特点是什么？

a、 组件是自包含的、可重用的软件单元；容器是包含其它组件的组件。它们被用作组件的框架。

b、主要有DOM模式和SAX模式。前者是基于树的方法，后者是基于事件的方法。DOM方法一次性将数据读入内存，从而具有随时可访问性，但是需要占用比较多的内存和时间，而SAX不具有随时可访问性，但是类似流媒体技术，可以比较快的实现对XML的访问，并且占用的内存和载入时间都比较短。

1. 请简述N层架构的演变历程，含义，以及其优点。

最初是主机系统或单层结构，数据访问、表示和业务逻辑在一个应用中，功能紧耦合，代码复用、代码可维护性、和代码的修改十分困难； 其次发展到C/S架构，将业务逻辑可以放在服务器端或者客户端，使得显示和逻辑的分离；之后数据层单独划分出来，成为表示层次、业务逻辑层和数据服务层三层架构；业务逻辑层的进一步根据应用分成多层次就成为N层结构。

N层的优点是：

功能松耦合，代码复用、系统可维护性、可扩充性都比较好。

1. 请叙述在Web开发中，Javabean的四个范围page, request, session, application的含义分别是什么，请给出比较详细地描述。

page：JavaBean只能在当前页面中使用。在JSP页面执行完毕后，该JavaBean将会被进行垃圾回收。

request：JavaBean在相临的两个页面中有效。

session：JavaBean在整个用户会话过程中都有效。

application：JavaBean在当前整个Web应用的范围内有效。

1. 请给出运用JSP和 Servlet的主要联系和区别

(1)JSP是以另外一种方式实现的Servlet，Servlet是JSP的早期版本，在JSP中，更加注重页面的表现，而在Servlet中则更注重业务逻辑的实现。

(2)当编写的页面显示效果比较复杂时，首选是JSP。或者在开发过程中，HTML代码经常发生变化，而Java代码则相对比较固定时，可以选择JSP。而我们在处理业务逻辑时，首选则是Servlet。

(3)同时，JSP只能处理浏览器的请求，而Servlet则可以处理一个客户端的应用程序请求。因此，Servlet加强了Web服务器的功能。

(4)JSP代码修改后自动重新编译；Servlet修改后需要重新启动服务器。

1. 根据EJB和JavaBeans的比较，填充下列的对比表

|  |  |
| --- | --- |
| EJB | JavaBeans |
| EJB用于服务端应用开发 |  |
| EJB构件是可部署的 |  |
| EJB支持使用部署描述符对EJB应用进行定制化 |  |
| EJB构件是分布式对象 |  |
| EJB构件对终端用户不可见 |  |

答案：

|  |  |
| --- | --- |
| EJB | JavaBeans |
| EJB用于服务端应用开发 | JavaBeans可以用于客户端应用开发 |
| EJB构件是可部署的 | JavaBeans构件是不可部署的 |
| EJB支持使用部署描述符对EJB应用进行定制化 | JavaBeans中对应用的定制化只能在开发阶段 |
| EJB构件是分布式对象 | JavaBeans构件不是分布式对象 |
| EJB构件对终端用户不可见 | 部分JavaBeans构件对终端用户可见 |

1. 名词解释：（1）JNDI；（2）JMS；（3）JTA；（4）JAF;（5）RMI

(1)JNDI: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；

(2)JMS：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；

(3)JTA：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；

(4)JAF: 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；

(5)RMI:Remote Method Interface,远程方法调用。

1. 设计分析题：

采用MVC设计一个简单的实现用户名以及密码验证和登录，并且根据不同的用户身份自动跳转到相应页面，如果用户名和密码不匹配则转到显示登录错误的页面。

请给出采用J2EE设计的时候，你会设计一些什么组件，并且各自承担MVC的什么角色。

Login.jsp，erro.jsp, success.jsp :视图，接收用户输入用户名和密码，并且返回结果，处于视图（V）位置。

配置文件和Servlet: 进行流程控制和返回结果的判断，转到相应的处理。处于控制（C）。

Javabean或者EJB： 进行密码匹配，封装用户信息，处于模型（M）位置。

# 第五部分 连接前后端的web服务

1. 什么是Web Service?它有哪些优点和缺点？并简单解释一下SOAP和WSDL。

Web Service就是为了使原来各孤立的站点之间的信息能够相互通信、共享而提出的一种接口。

Web Service所使用的是Internet上统一、开放的标准，如HTTP、XML、SOAP（简单对象访问协议）、WSDL等，所以Web Service可以在任何支持这些标准的环境（Windows,Linux）中使用。

优势：

(1).跨平台。

(2).SOAP协议是基于XML和HTTP这些业界的标准的，得到了所有的重要公司的支持。

(3).由于使用了SOAP，数据是以ASCII文本的方式而非二进制传输，调试很方便；并且由于这样，它的数据容易通过防火墙，不需要防火墙为了程序而单独开一个“漏洞”。

(4).此外，WebService实现的技术难度要比CORBA和DCOM小得多。

(5).要实现B2B集成，EDI比较完善与比较复杂；而用WebService则可以低成本的实现，小公司也可以用上。

(6).在C/S的程序中，WebService可以实现网页无整体刷新的与服务器打交道并取数。

缺点：

(1).WebService使用了XML对数据封装，会造成大量的数据要在网络中传输。

(2).WebService规范没有规定任何与实现相关的细节，包括对象模型、编程语言，这一点，它不如CORBA。

SOAP协议（Simple Object Access Protocal,简单对象访问协议）,它是一个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。在此协议下，软件组件或应用程序能够通过标准的HTTP协议进行通讯。它的设计目标就是简单性和扩展性，这有助于大量异构程序和平台之间的互操作性，从而使存在的应用程序能够被广泛的用户访问。

 WSDL（Web Service Description Language）是采用XML语言来描述Web Service的属性的语言。WSDL文档可以包含以下内容：What：Web Service做什么；Where：Web Service位于哪里；How：怎样调用。如果将Web Service作为一个分布式对象来看，WSDL就是Web Service的接口描述语言（IDL）。WSDL定义了一套基于XML的语法，将Web Service描述为能够进行消息交换的服务访问点的集合。

1. 请详细说明Web Service构架中的三种角色和三种基本操作。

Web Service构架中的三种角色分别是服务提供者、服务请求者以及服务注册中心。

三个基本操作：

（1）发布：服务提供者定义 Web 服务的服务描述并把它发布到服务请求者或服务注册中心

（2）查找：服务请求者使用查找操作来从本地或服务注册中心检索服务描述

（3）绑定/调用：使用服务描述与服务提供者进行绑定并调用 Web 服务实现或同它交互

1. 请说明SOA的含义，以及SOA与Web Services的关系。

面向服务的架构。是交错的概念。Web Services目前是实现SOA的一种最常用的技术。

1. 请简单说明以下较新的Web技术的要点和你的理解，给出可能的应用场景：REST化的Web架构。

分布式超媒体系统设计的一种架构风格：

网络上的所有事物都被抽象为资源（resource）；

每个资源对应一个唯一的资源标识符（resource identifier）；

通过通用的连接器接口（generic connector interface）对资源进行操作；

对资源的各种操作不会改变资源标识符；

所有的操作都是无状态的（stateless）。

基于Http协议的，任何对资源的操作行为都是通过Http协议来实现

1. 概述中间件的概念，以及几类中间件的主要特征

中间件是独立于应用程序的软件，它提供服务用来协调不同的应用软件。

五类典型的中间件：

远程过程调用

数据库访问的中间件

面向消息的中间件

分布式对象技术

事务处理监控器

1. 离线Web应用

在本地有轻量级服务器，甚至数据库；使得网络断掉的情况下也可以采用web客户端工作，在线会自动同步。

1. 依据下图所示，给出web服务的SOAP调用过程描述；并说明SOAP服务器如Axis处于什么位置，其工作原理是什么？



（1）服务请求者的应用程序创建一条 SOAP 消息，将此信息和服务提供者的网址一起提供给 SOAP 基础结构（例如一个 SOAP 客户机运行时）。SOAP 客户机运行时与一个底层网络协议（例如 HTTP）交互，然后在网络上将 SOAP 消息发送出去。

（2）网络基础结构将消息传送到服务提供者的 SOAP 运行时（例如一个 SOAP 服务器）。SOAP 服务器将请求消息路由到服务提供者的 Web 服务。如果应用程序需要，SOAP 运行时负责将 XML 消息转换为特定于编程语言的对象。这个转换由消息中可以找到的编码模式所控制。

（3） Web 服务负责处理请求信息并生成一个响应。该响应也是一条 SOAP 消息。响应的 SOAP 消息）被提供给 SOAP 运行时，其目的地是服务请求者。在 HTTP 上的同步请求／响应的情况中，联网协议的底层请求／响应本质用于实现消息传递的请求／响应本质。SOAP 运行时将 SOAP 消息响应发送到网络上的服务请求者。

（4）响应消息由服务请求者节点上的联网基础结构接收。消息会经过整个 SOAP 基础结构；可能会将 XML 消息转换为目标编程语言中的对象。然后，响应消息被提供给应用程序

SOAP服务器属于SOAP运行时，对SOAP消息进行解析和封装，并映射到后端的web服务功能上。典型的Axis就是一个部署在容器中的web组件。

1. 根据下图回答问题：



(a)在如图简化的订单处理中，哪些系统和功能可以封装为web服务，封装为web服务的优点是什么？（10分）

(b) 要整合这些服务，可以采用BPEL标准。以该订单处理系统为例，阐述BPEL是如何整合web服务的。（10分）

（a）信用卡检测，库存管理，以及前端的订单处理系统都可以封装为web Services。优点是集成了已有的遗留系统，并且易于实现自动流程。

（b）



1. Web Service不适用的场景是什么？

1、单机应用程序

目前，企业和个人还使用着很多桌面应用程序。其中一些只需要与本机上的其它程序通信。在这种情况下，最好就不要用WebService，只要用本地的 API就可以了。COM非常适合于在这种情况下工作，因为它既小又快。运行在同一台服务器上的服务器软件也是这样。最好直接用COM或其它本地的API来 进行应用程序间的调用。当然WebService也能用在这些场合，但那样不仅消耗太大，而且不会带来任何好处。

2、局域网的同构应用程序

在许多应用中，所有的程序都是用VB或VC开发的，都在Windows平台下使用COM，都运行在同一个局域网上。例如，有两个服务器应用程序需要相互通 信，或者有一个Win32或WinForm的客户程序要连接局域网上另一个服务器的程序。在这些程序里，使用DCOM会比SOAP/HTTP有效得多。与 此相类似，如果一个.NET程序要连接到局域网上的另一个.NET程序，应该使用.NETremoting。有趣的是，在.NETremoting 中，也可以指定使用SOAP/HTTP来进行WebService调用。不过最好还是直接通过TCP进行RPC调用，那样会有效得多。

1. 怎么理解UDDI？

UDDI是Universal Description Discovery and Integration的缩写，即统一描述、发现和整合规范。用来注册和查找服务，把web services收集和存储起来，这样当别人访问这些信息的时候就从UDDI中查找，看有没有这个信息存在。

1. WSDL文档主要有那几部分组成，分别有什么作用？

一个WSDL文档的根元素是definitions元素，WSDL文档包含7个重要的元素：types, import, message, portType, operations, binding和service元素。

1.definitions元素中一般包括若干个XML命名空间；

2.Types元素用作一个容器，定义了自定义的特殊数据类型，在声明消息部分（有效负载）的时候，messages定义使用了types元素中定义的数据类型与元素；

3.Import元素可以让当前的文档使用其他WSDL文档中指定命名空间中的定义；

4.Message元素描述了Web服务的有效负载。相当于函数调用中的参数和返回值；

5.PortType元素定义了Web服务的抽象接口，它可以由一个或者多个operation元素，每个operation元素定义了一个RPC样式或者文档样式的Web服务方法；

6.Operation元素要用一个或者多个messages消息来定义它的输入、输出以及错误；

7.Binding元素将一个抽象的portType映射到一组具体的协议（SOAP或者HTTP）、消息传递样式（RPC或者document）以及编码样式（literal或者SOAP encoding）；

8.Service元素包含一个或者多个Port元素

每一个Port元素对应一个不同的Web服务，port将一个URL赋予一个特定的binding，通过location实现。可以使两个或者多个port元素将不同的URL赋给相同的binding。