### Java ee 概述：

##### JavaEE 的概念：

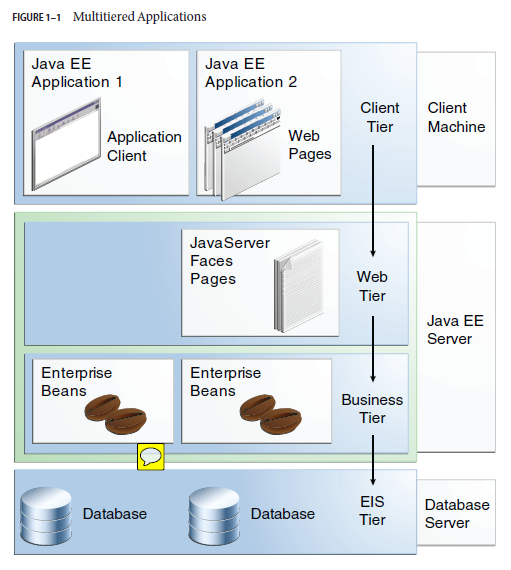
* Java Platform, Enterprise Edition (Java EE) 6 is the industry standard for enterprise Java computing.
* Utilize the new, lightweight Java EE 6 Web Profile to create next-generation web applications, and the full power of the Java EE 6 platform for enterprise applications.
* Developers will benefit from productivity improvements with more annotations, more POJOs, simplified packaging, and less XML configuration.

##### 分布式多层，包含哪些层？每一层有哪些组件？

The Java EE platform uses a distributed multitiered application model for enterprise applications.

Enterprise Information System Tier：

* The enterprise information system tier handles EIS software and includes enterprise infrastructure systems such as enterprise resource planning (ERP), mainframe transaction processing, database systems, and other legacy information systems.



#### 服务器以容器的形式提供服务，都有哪些容器以及提供哪些服务？

Container Types

* Java EE server: The runtime portion of a Java EE product. A Java EE server provides EJB and web containers.

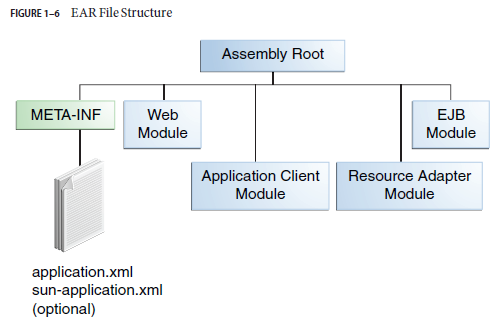
Java EE server, including services such as security, transaction management, JavaNaming and Directory Interface (JNDI) lookups, and remote connectivity.

* + Enterprise JavaBeans (EJB) container: Manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.
  + Web container: Manages the execution of web pages, servlets, and some EJB components for Java EE applications. Web components and their container run on the Java EE server.
* Application client container: Manages the execution of application client components. Application clients and their container run on the client.
* Applet container: Manages the execution of applets. Consists of a web browser and Java Plug-in running on the client together.

Java ee 的应用打包部署，每个单元包含哪些内容？

A Java EE application is delivered in either a Java Archive (JAR) file, a Web Archive (WAR) file, or an Enterprise Archive (EAR) file.

deployment descriptors ：There are two types of deployment descriptors: Java EE and runtime.



#### Java ee 包含哪些模块？模块的内容是什么？每个模块的扩展名是什么？

Java EE module The four types of Java EE modules

* + EJB modules, which contain class files for enterprise beans and an EJB deployment descriptor. \*.jar
  + Web modules, which contain servlet class files, web files, supporting class files, GIF and HTML files, and a web application deployment descriptor. \*.war
  + Application client modules, which contain class files and an application client deployment descriptor. \*.jar
  + Resource adapter modules, which contain all Java interfaces, classes, native libraries, and other documentation, along with the resource adapter deployment descriptor. \*.rar

#### Java EE应用可重用模型，哪些角色？

Development Roles

* The Java EE product provider
  + designs and makes available for purchase the Java EE platform APIs, and other features defined in the Java EE specification.
  + typically application server vendors who implement the Java EE platform according to the Java EE Platform specification.
* The tool provider
  + creates development, assembly, and packaging tools used by component providers, assemblers, and deployers.
* Application Component Provider
  + creates web components, enterprise beans, applets, or application clients for use in Java EE applications.
* Application Assembler
  + receives application modules from component providers and assembles them into a Java EE application EAR file.
* Application Deployer and Administrator
  + configures and deploys the Java EE application, administers the computing and networking infrastructure where Java EE applications run, and oversees the runtime environment.

#### 广泛使用的web 容器

* Web服务器：Tomcat, Resin
* EJB容器： JBoss
* 应用服务器：IBM Websphere, Oracle WebLogic,

### 关于Web应用的概述

#### Web应用是Web应用服务器的动态拓展，哪两种类型？

* A *web application* is a dynamic extension of a web or application server. There are two types of web applications:
  + Presentation-oriented: A presentation-oriented web application generates interactive web pages containing various types of markup language (HTML, XHTML, XML, and so on) and dynamic content in response to requests.
  + Service-oriented: A service-oriented web application implements the endpoint of a web service.

#### 一个JavaWeb应用的请求处理流程。

* The client sends an HTTP request to the web server.
* A web server that implements Java Servlet and JavaServer Pages technology converts the request into an HTTPServletRequest object.
* This object is delivered to a web component, which can interact with JavaBeans components or a database to generate dynamic content.
* The web component can then generate an HTTPServletResponse or can pass the request to another web component.
* A web component eventually generates a HTTPServletResponse object.
* The web server converts this object to an HTTP response and returns it to the client.

#### 一个Web应用会包括哪些内容？

* A web application consists of web components, static resource files such as images, and helper classes and libraries.
* Web container

#### Web组件可以是Servlet和web页面，哪些技术可以实现Web页面（JSP, JavaServer Faces and Facelets pages）

#### Web应用的一些特征可以在部署到容器是被配置，两种配置的方式是什么？ （Java EE annotations， deployment descriptor (DD)）

#### Web容器提供了哪些服务？

provides such services as request dispatching, security, concurrency, and lifecycle management

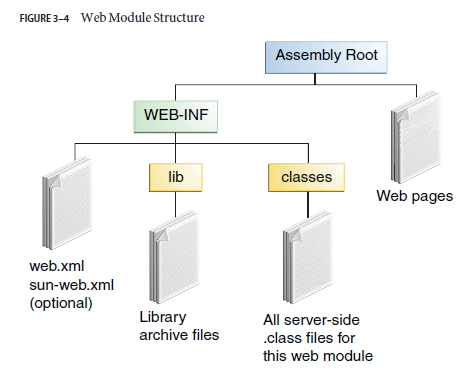
#### Web模块的目录结构

* The top-level directory of a web module is the *document root of the application.*
  + The document root is where XHTML pages, client-side classes and archives, and static web resources, such as images, are stored.
* The document root contains a subdirectory named WEB-INF.

WEB-INF contains the following files and directories:

* + classes: A directory that contains *server-side classes: servlets, enterprise bean class files, utility classes, and JavaBeans* components
  + tags: A directory that contains tag files, which are implementations of tag libraries
  + lib: A directory that contains JAR files that contain enterprise beans, and JAR archives of libraries called by server-side classes

Deployment descriptors, such as web.xml (the web application deployment descriptor) and ejb-jar.xml (an EJB deployment descriptor)



### Servlet

Servlet 生命周期。

* **The life cycle of a servlet is controlled by the container in which the servlet has been deployed.**
* **When a request is mapped to a servlet, the container performs the following steps.**
  + **1. If an instance of the servlet does not exist, the web container**
    - **a. Loads the servlet class.**
    - **b. Creates an instance of the servlet class.**
    - **c. Initializes the servlet instance by calling the init method.**
  + **2. Invokes the service method, passing request and response objects.**
* **If the container needs to remove the servlet, it finalizes the servlet by calling the servlet’s destroy method.**

Http请求，URL组成部分包括哪些元素？

**http://[*host*]:[*port*][*request-path*]?[*query-string*]**

**The request path is further composed of the following elements:**

* + **Context path: A concatenation of a forward slash (/) with the context root of the servlet’s web application.**
  + **Servlet path: The path section that corresponds to the component alias that activated this request. This path starts with a forward slash (/).**
  + **Path info: The part of the request path that is not part of the context path or the servlet path.**

Tomcat的组件结构，以及如何处理请求。

* **Server：代表一个服务器**
* **Connector：在某个指定端口上侦听客户请求，并将获得的请求交给Engine来处理**
  + **Java HTTP Connector在端口8080侦听来自客户Browser的http请求**
  + **AJP 1.3 Connector在端口8009侦听来自其他Web Server(Apache)的JSP/Servlet代理请求**
* **Engine：将获得的请求匹配到某个虚拟主机(Virtual Host)上，并把请求交给该Host来处理**
* **Host：代表虚拟主机，每一个都和某个网络域名相匹配，每一个都可部署多个Web应用(对应不同的Context)**
* **Context：对应一个Web应用（由一些Servlet、HTML页面、Java类、JSP页面和一些其他的资源构成）。**

**Tomcat处理http请求**

* [**http://localhost:8080/HelloWorld/**](http://localhost:8080/HelloWorld/)
* **1.请求被发送到本机端口8080，被Java HTTP Connector获得；**
* **2. Connector将该请求交给它所在的Service的Engine来处理，并等待Engine的回应；**
* **3. Engine获得请求，匹配所有虚拟主机；**
* **4. Engine匹配到名为localhost的主机；**
* **5. localhost主机获得请求，匹配所拥有的所有Context；**
* **6. localhost主机匹配到路径为/HelloWorld的Context**
* **7.路径为/HelloWorld的Context获得请求，在映射表中寻找对应的Servlet；**
* **8.Context匹配到URL PATTERN为/的Servlet;**
* **9.构造HttpServletRequest对象和HttpServletResponse对象，作为参数调用该Servlet的Service方法；**
* **10. Context把执行完之后的HttpServletResponse对象返回给localhost主机；**
* **11.Host把HttpServletResponse对象返回给Engine；**
* **12. Engine把HttpServletResponse对象返回给Connector；**
* **13. Connector把HttpServletResponse对象返回给客户Browser。**

Session 的两种实现机制（详细了解）

**Sessions are represented by an HttpSession object. You access a session by calling the getSession method of a request object. This method returns the current session associated with this request, or, if the request does not have a session, it creates one.**

* **从HttpServletRequest对象中获得会话对象**
  + **HttpSession session=request.getSession(true);**
  + **true：如果会话对象不存在，一个新的会话对象将被创建——如果不断提交请求，一个人可以创建大量的会话对象，耗尽服务器上的内存资源**
* **好的实践：**
  + **HttpSession session=request.getSession(false);**
  + **//如果会话对象不存在，不能创建新会话**
  + **if(session==null){**
    - **response.sendRedirect(“http://……/login”);**
    - **//强迫用户登录，成功登录之后，才能创建新会话**
    - **//当用户注销时，使会话失效--session.invalidate()**
  + **}**

Web组件可以调用其他的外部资源，采用什么方式调用？（包含页面进来）

**Web components can invoke other web resources in two ways: indirectly and directly:**

**1 A web component indirectly invokes another web resource when it embeds a URL that points to another web component in content returned to a client.**

**2 A web component can also directly invoke another resource while it is executing. There are two possibilities: The web component can include the content of another resource, or it can forward a request to another resource.**

Session和Cookie 使用场景分别是什么？

**Session**:**跟踪用户的购物车** **导航信息，登录状态**

**Cookie** :**用户登录ID** **用户对语言和颜色的选择之类的偏好**, **跟踪应用程序的使用情况**,**cookie.txt文件**

Filter使用场景是什么？（在什么情况下使用）

**(The main tasks that a filter can perform are as follows:**

* + **Query the request and act accordingly.**
  + **Block the request-and-response pair from passing any further.**
  + **Modify the request headers and data.**
  + **Modify the response headers and data.**
  + **Interact with external resources.)**
* **改善代码重用，在不修改servlet代码的情况下向servlet添加功能；**
  + **如：身份认证**
* **可用于跨多个servlet执行一些功能，创建可重复使用的功能**
* **在servlet处理请求之前，截获请求** **如：在调用servlet之前，截获请求，验证用户身份，未经授权的用户遭到拒绝，而servlet不知道曾经有过这样的请求**
* **(过滤器的应用：代码重用 应用安全策略 日志 为特定目标浏览器传输XML输出 图像转换、加密 动态压缩输出 )**

Listener的使用场景？

* **You can monitor and react to events in a servlet’s life cycle by defining listener objects whose methods get invoked when life-cycle events occur.**

如何处理Servlet的线程安全问题？

**servlet默认是多线程的，Server创建一个实例，用它处理并发请求——编写线程安全的类，避免使用可以修改的类变量和实例变量；**

**修改servlet类，实现SingleThreadMode1接口——单线程**

### JSP

JSP生命周期:

* **When a request is mapped to a JSP page, the web container first checks whether the JSP page’s servlet is older than the JSP page. If the servlet is older, the web container translates the JSP page into a servlet class and compiles the class. During development, one of the advantages of JSP pages over servlets is that the build process is performed automatically.**

JSP是文本文档，包括哪两种类型的文本？

**contains two types of text: static data, JSP elements**

JSP页面有两种语法表达方式，哪两种？**: standard and XML**

JSP元素都有哪些？

Directives,scripting elements,expression language expressions,jsp:[set|get],jsp:[include|forward],jsp:plugin, **custom tags**

在JSP页面的翻译阶段，要去处理构建动态内容的JSP元素，是如何处理这些元素的？（指令、脚本、动作（include,forward）在翻译是如何处理？定制标签如何处理？）

* **During the translation phase each type of data in a JSP page is treated differently.** 
  + **Static data is transformed into code that will emit the data into the response stream.**
  + **JSP elements are treated as follows:**
    - **Directives are used to control how the web container translates and executes the JSP page.**
    - **Scripting elements are inserted into the JSP page’s servlet class.**
    - **Expression language expressions are passed as parameters to calls to the JSP expression evaluator.**
    - **jsp:[set|get]Property elements are converted into method calls to JavaBeans components.**
    - **jsp:[include|forward]elements are converted into invocations of the Java Servlet API.**
    - **The jsp:plugin element is converted into browser-specific markup for activating an applet.**
    - **Custom tags are converted into calls to the tag handler that implements the custom tag.**

有哪些常用的JSP指令？

|  |  |  |
| --- | --- | --- |
|  | **XML语法** | **标准语法** |
| **page指令** | **<jsp:directive.page attribute list/>** | **<%@ page property-attribs %>** |
| **include指令** | **<jsp:directive.include file="filename"/>** | **<%@ include file="filename" %>** |
| **Taglib指令** | **xmlns:prefix="tag library URL"** | **<%@ taglib uri="uri" prefix="tagPrefix" %>** |

脚本元素中创建和使用对象的方法有哪些？这些对象在翻译时如何进行处理？

* **There are three ways to create and use objects in scripting elements:**
  + **Instance and class variables of the JSP page’s servlet class are created in *declarations* and accessed in *scriptlets* and *expressions*.**
  + **Local variables of the JSP page’s servlet class are created and used in *scriptlets* and *expressions*.**
  + **Attributes of scope objects are created and used in *scriptlets* and *expressions*.**

JSP动作的作用是改善脚本、拜托Java代码，有哪些常用动作？

**标准动作：** **<jsp:include>** **<jsp:forward>** **<jsp:plugin>** **<jsp:useBean>, < jsp:setProperty >, < jsp:getProperty >**

Include动作和Include指令的区别？

**include指令：包括其他页面，编译时把其他页面的内容加进来，比include动作快**

**include标准动作：使用RequestDispatcher，运行时把其他页面的内容加进来（包括到输出流中）**

**被包含的资源为变量时：**

* + **<jsp:include page=** " **%=dynamicRef %**" **/>**
    - **使用了表达式**

Forward 动作 和 Redirect 动作的区别？

* **把当前JSP请求转发到另一个资源上**
* **<jsp:forward page=** " **foo.jsp**" **/>**
* **与HTTP重定向的差别：**
  + **HTTP重定向：response.sendRedirect(*myNewURL*);**
    - **发送的请求信息又回送给客户机，让客户机再转发到另一个资源上，新的URL出现在Web浏览器中，需要在服务器和客户机之间增加一次通信**
  + **forward标准动作：使用RequestDispatcher ，JSP的转发功能是在服务器本身上实现的**

JSP有四种作用域对象，分别是什么？

**pageScope, requestScope, sessionScope, applicationScope**

5. JDBC

两种建立数据库连接的方式？

**JDBC-ODBC桥 Native-API Partly Java**

事务：哪些种Java事务类型？

**JDBC事务、JTA(Java Transaction API)事务、容器事务**

什么是JDBC？（简单解释）

**Java数据库连接（Java DataBase Connectivity）**

**ODBC——开放式数据库连接**

**使用针对于数据库的驱动程序，提供一组用于访问任何数据库的API**

什么是JTA？

* **The Java EE architecture provides a default auto commit to handle transaction commits and rollbacks.**
* **An *auto commit* means that any other applications that are viewing data will see the updated data after each database read or write operation.**
* **However, if your application performs two separate database access operations that depend on each other, you will want to use the JTA API to demarcate where the entire transaction, including both operations, begins, rolls back, and commits.**

### MVC

基于MVC模式的JavaEE Web应用构架:

* **由一个servlet应答初始的请求；**
* **Servlet完成实际的数据处理并将结果存储在bean中；**
* **Bean存储在HttpServletRequest, HttpSession, 或ServletContext中；**
* **Servlet使用RequestDispatcher的forward方法将请求转发到JSP页面；**
* **JSP页面通过使用jsp:useBean和相应的作用域（request, session, 或application ）从bean中读出数据。**

常用的MVC框架？使用框架的优点：

**把多个组件集成到一起，相互合作，协调一致的进行工作**

* + **模型：封装应用数据（关系数据库或EJB），处理商业逻辑**
  + **视图：呈现给用户的界面（JSP或应用GUI）**
  + **控制器：接受用户动作，并对应用数据进行适当的处理（Servlet）**
* **The MVC architecture allows you to separate the view from the model via a controller.**
* **The MVC design allows for concurrent development of each component comprising the model, view and controller by different programmers**
* **Ability to react to changes:**

**Can change the user interface without impacting business logic, for example, change the text field to a list of choices.**

**Can change the business logic without impacting the UI, for example, tax laws change every year, but UI is still amount owed or refunded.**

**Can move location of data without impacting the UI or business logic, for example, migrate from test system to production.**

**Make it easy to incorporate new technologies**

**Be able to support different types of clients**

基于JavaEE 做应用时，有哪些常用的设计模式？**前端控制器模式**

### 消息服务JMS

使用消息的应用场景:

**电子商务应用:进程中的通信**

两种消息应用：队列、主题，使用场景是什么:

队列:**点对点消息处理 主题：发布/订阅消息处理（pub/sub）**

对于消息的消费者来说：同步、异步，应用场景是什么:

**异步通信：不必等到所有的处理工作都完成，成功将消息插入队列后，即可恢复处理**

**同步：如果消息未到达，消息消费者将一直等待**

**Synchronously: A subscriber or a receiver explicitly fetches the message from the destination by calling the receive method. The receive method can block until a message arrives or can time out if a message does not arrive within a specified time limit.**

**Asynchronously: A client can register a *message listener* with a consumer. A message listener is similar to an event listener. Whenever a message arrives at the destination, the JMS provider delivers the message by calling the listener’s onMessage method, which acts on the contents of the message.**

### 中间件

中间件技术的类型：

* **远程过程调用（RPC）** **远程数据库访问** **分布式事务处理** **消息队列**

名词解释：RMI、JNDI：

* **Java Remote Method Invocation over the Internet Inter-ORB Protocol**
* **实现强大的网络功能的机制**
* **允许编写分布式对象，可以使对象在内存中、跨Java虚拟机和跨物理设备进行通信**
* **RMI是用于分布式编程的、基于Java的模型；**
* **使用RMI，Java代码可以调用远程对象中的方法；**
* **调用远程对象：把请求传递给远程对象，执行方法调用，然后传回方法的返回值**

**;使用java.rmi包开发RMI应用程序**;**RMI应用程序有两个组件：RMI服务器**,**RMI客户**

**JNDI：Java命名和目录接口(Java Naming and Directory Interface)，为开发人员提供了查找和访问各种命名和目录服务的通用、统一的方式。（中央注册中心，储存了各种对象、用户和应用的变量及其值，开发大型的分布式应用，使分布式的Java程序找到分布式的对象）**

### EJB

最新的JavaEE规范中，有哪两种企业Bean？:

**Session, Message-driven**

会话Bean的三种类型:

**Session beans are of three types: stateful, stateless, and singleton**

EJB的三种客户类型:

* + **Remote** **Local** **web service**

客户可以以两种方式获得对企业Bean的引用，哪两种？:

**Clients access enterprise beans either through a *no-interface view or through a business interface.***

企业Bean包括哪些内容？目录结构是什么？:

* **Enterprise bean class:**
  + **Implements the business methods of the enterprise bean and any lifecycle callback methods.**
* **Business interfaces:**
  + **Define the business methods implemented by the enterprise bean class.**
  + **A business interface is not required if the enterprise bean exposes a local, no-interface view.**
* **Helper classes:**
  + **Other classes needed by the enterprise bean class, such as exception and utility classes.**

### JPA

持久的概念: 对象关系映射的概念。

* **Java对象序列化：将对象序列化，应用于网络通信或简单的持久存储**
* **对象/关系型数据库映射：使用关系型数据库持久存储Java对象（利用JDBC ）**