

# 第2章 软件体系结构风格

## 补充内容--JavaEE Tutorial

2.1 分布式多层应用系统

2.2 JavaEE Containers

2.3 JavaEE Application Assembly and Deployment

2.4 Packaging Applications

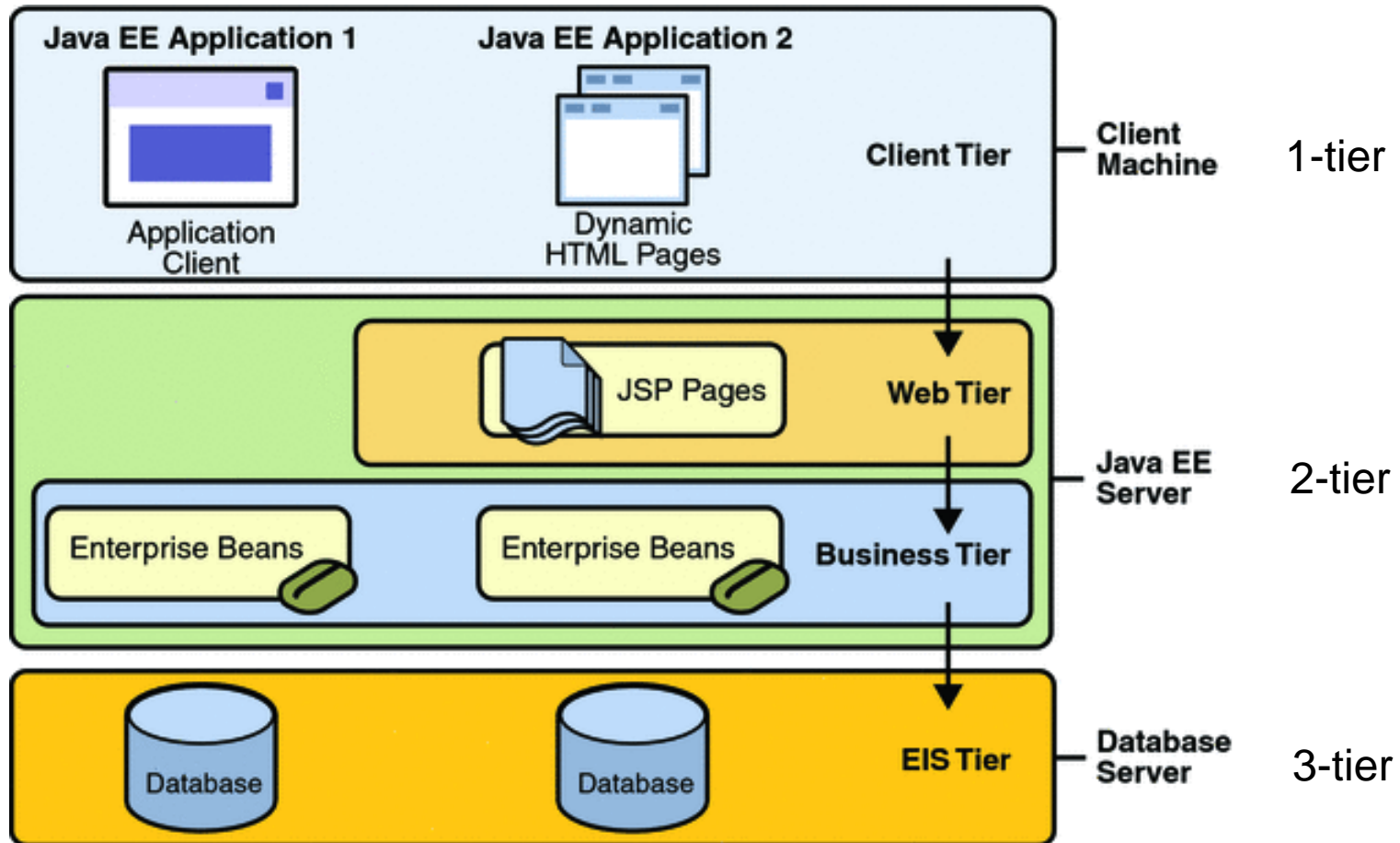
2.5 case:Library Application

内容来源于Sun <<JavaEE tutorial>>

## 2.1 分布式多层应用系统.

1) JavaEE 平台是**分布式多层**应用系统模型(distributed multitiered application model)

下图图示了两个JavaEE应用系统被划分成多层的例子.



**Figure 2-1 Multitiered Applications**

2)应用系统模型依据系统功能的不同，逻辑上把应用系统划分成各个组件。

- 这些组件共同完成系统的功能。
- 依据组件完成系统功能的差异，这些组件被部署在**JavaEE**平台的不同层上。
- 不同层的组件可安装在不同的机器上。

- Client-tier components run on the client machine.
- Web-tier components run on the JavaEE server.
- Business-tier components run on the JavaEE server.
- Enterprise information system (EIS)-tier software runs on the EIS server.

在图中，JavaEE应用系统虽然由三层或者四层构成。但通常来说，根据系统分布在不同的机器上，JavaEE应用系统被认为由 3 层构成：

- client machines,
- JavaEE server machine,
- database

三层应用系统模型扩展了典型的两层应用系统模型（ client and server model ）：

- client and server model: 应用系统部署在 the client application and back-end storage.
- Three-tiered applications model: placing a **multithreaded application server** between the client application and database

## 2.1.2 JavaEE Components

### 1) JavaEE Components 概述

(1)JavaEE应用系统是由组件构成的.

(2)一个**JavaEE组件**是一个包含有一定功能的软件单元, 它是由一些相关的文件和类构成.

(3) 所有的组件组合在一起构成应用系统, 组件与组件之间可以相互通信.

JavaEE规范定义了以下一些JavaEE组件:

- Application clients and applets: components
  - run on the client
- Java Servlet, JavaServer Faces, and JavaServer Pages (JSP ) : Web components.
  - run on the server.
- Enterprise JavaBeans(EJB) components (enterprise beans) : business components.
  - run on the server.

**JavaEE组件是用java编写的，并且编译过程与其它java程序一样。**

**JavaEE组件与标准的java类不同之处在于：**

**JavaEE组件被打包(assembled)成JavaEE应用系统。**

**JavaEE组件需要符合JavaEE规范。**

**JavaEE组件被部署在服务器上，在JavaEE服务器中运行以及由JavaEE服务器管理**

## 2) JavaEE Clients

A JavaEE client can be a **Web client** or an **application client**.

### Web Clients:

一个 Web client由两部分构成:

(1)动态web页面:各种标记语言 (HTML, XML等等)

由运行在Web tier上的Web components产生的.

(2)一个Web browser:显示从服务器上接受到的页面.

thin client:不执行heavyweight operations (EJB完成)

# Application Clients

- 1) runs on the client machine
- 2) 一般是GUI( graphical user interface )，但命令行式的Application Clients不需要GUI.
- 3) 可以直接访问enterprise beans  
注明：如果有安全方面的需求，JavaEE客户端可以通过servlet访问EJB
- 4 ) 可以是non-Java languages(JavaEE平台也可以与之交互).



### 3)JavaBeans Component

- ①Server tier和client tier可以包含JavaBeans component
- ②JavaBeans component: 可以管理application client 与 components running on the JavaEE server 间的数据流, 也可以管理server components与 database间的数据流.
- ③JavaBeans components are not considered JavaEE components by the JavaEE specification.
- ④naming and design conventions : get和set方法.

## 4 ) JavaEE Server Communications

[Figure 2-2](#)说明了可构成client tier的各种元素。

1 ) Client可以直接与JavaEE server 的business tier 通信。

2 ) Client (运行在browser中)可以通过运行在Web tier中的jsp或servlet间接与JavaEE server 的business tier 通讯。

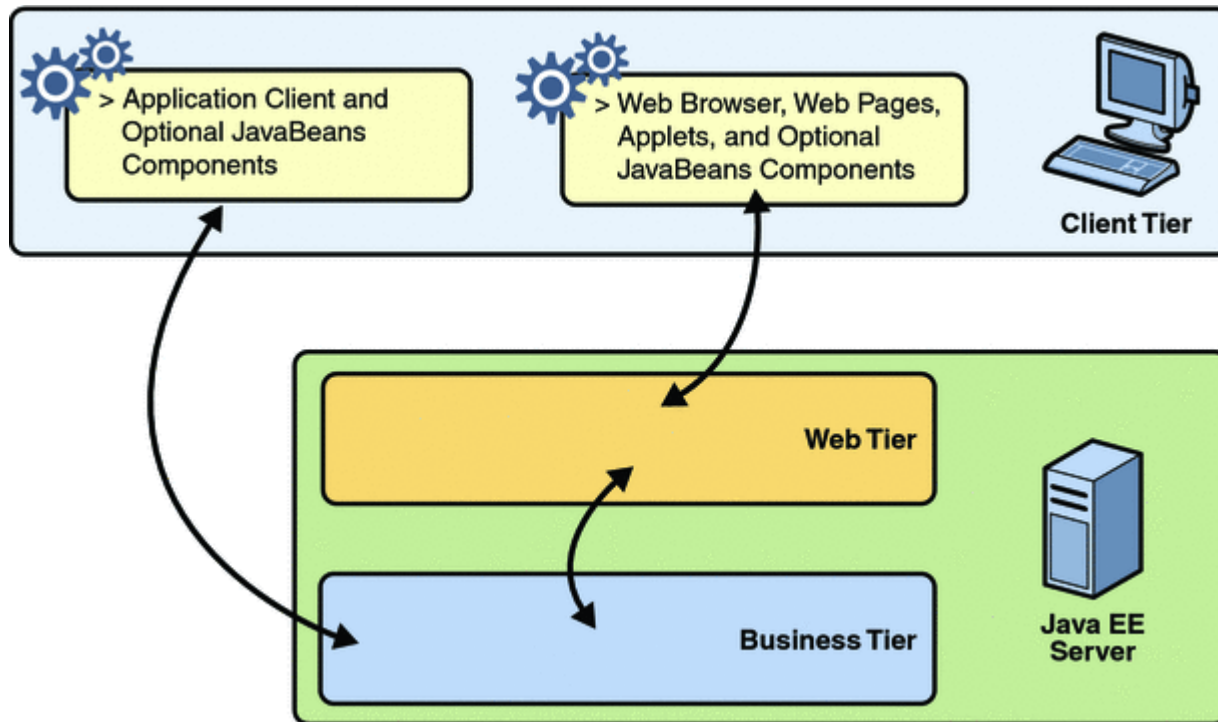


Figure 2-2 Server Communications

## 5 ) Web Components

JavaEE Web components可以是servlets 、 JSP pages、 JavaServer Faces.

- (1)Servlets是java语言编写的类,它可以动态地处理请求(requests)和创建响应(responses).
- (2) JSP pages are text-based documents  
JSP pages execute as servlets  
JSP pages allow a more natural approach to creating static content.
- (3)JavaServer Faces technology builds on servlets and JSP technology and provides a user interface component framework for web applications.

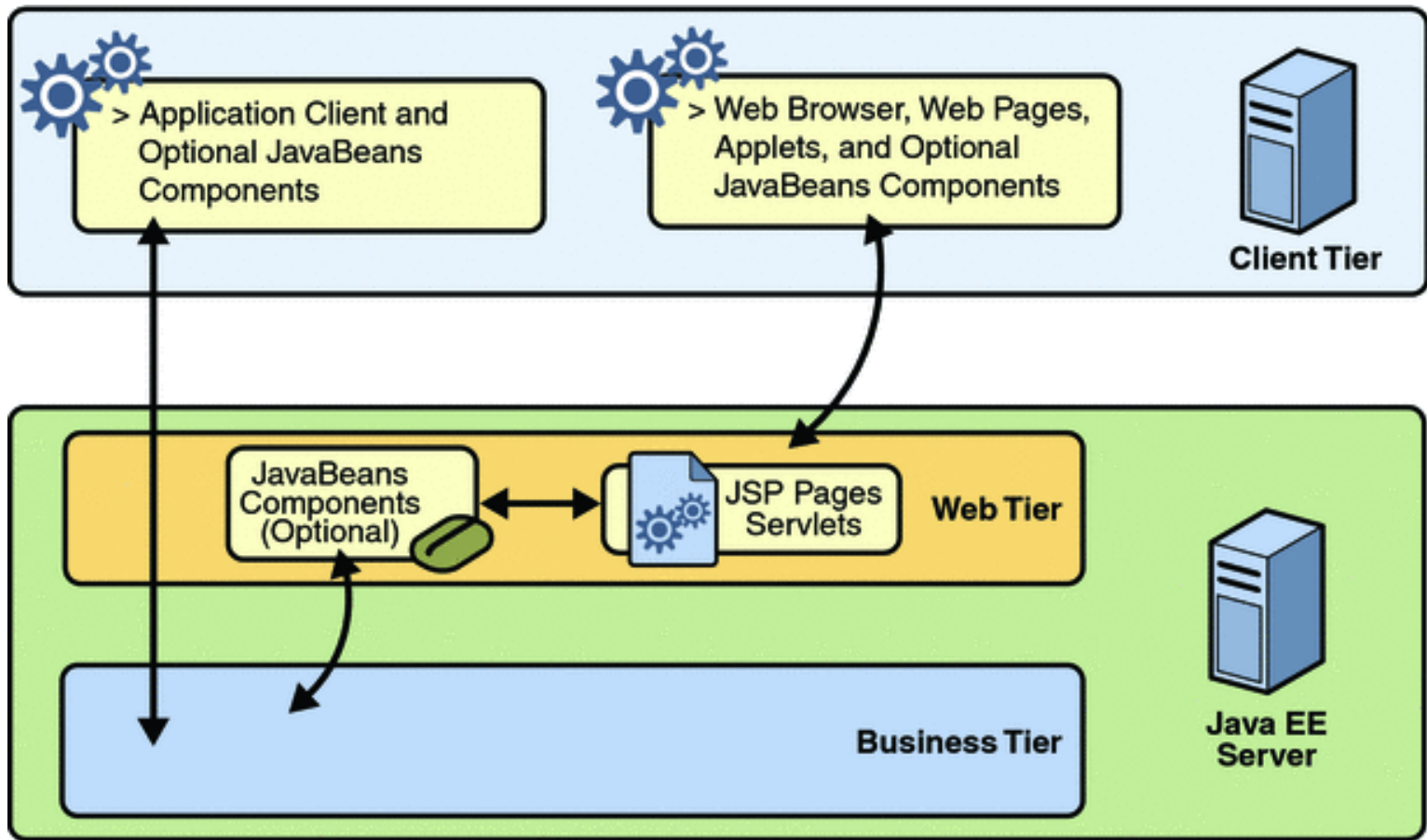


Figure 2-3 Web Tier and JavaEE Application

说明:

在系统assembly 的时候，Static HTML pages 、 applets、 Server-side utility classes是和Web components 捆绑在一起的，但是JavaEE specification认为它们不是Web components 的一部分。

HelloWorldServlet.java

HelloWorldJSP.jsp

## 6 ) Business Components

- (1) Business code 是用来解决和处理部分商业领域 (banking, retail, or finance) 需求的 logic.
- (2) Business code is handled by **enterprise beans** running in the business tier.

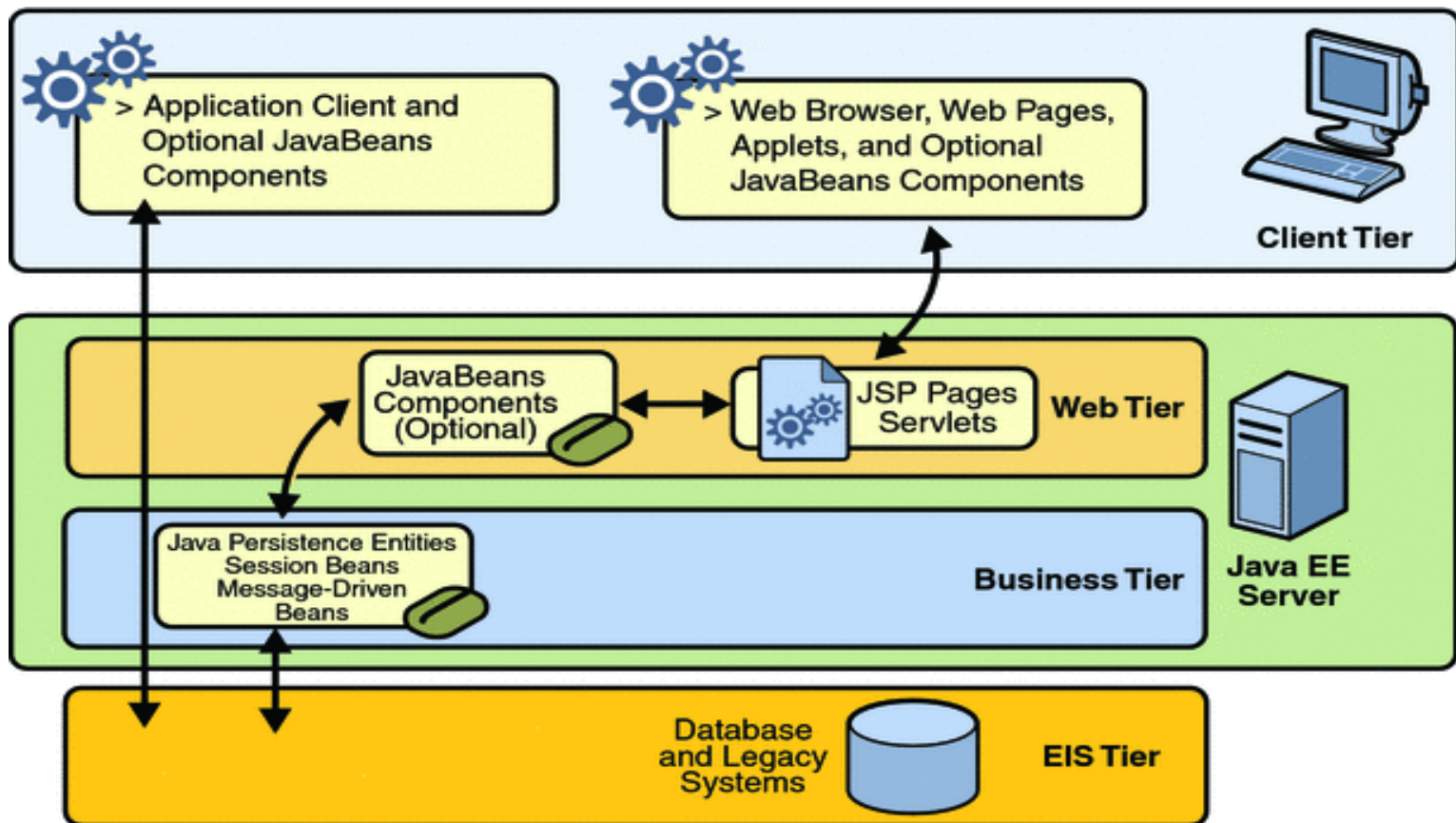


Figure 2-4 Business and EIS Tiers

(3) 三种类型的enterprise beans : **session beans**, **entity beans**, and **message-driven beans**.

- A *session bean* represents a transient conversation with a client.

When the client finishes executing, the session bean and its data are gone.

- an *entity bean* represents persistent data stored in one row of a database table.

If the client terminates or if the server shuts down, the underlying services ensure that the entity bean data is saved.



A *message-driven bean* combines features of a session bean and a Java Message Service (“JMS”) message listener, allowing a business component to receive JMS messages asynchronously (异步).

说明:

- (1) entity beans 被 **Java persistence API entities** 替代.
- (2) **An entity** represents **persistent data** stored in one row of a **database table**.

If the client terminates, or if the server shuts down, the persistence manager ensures that the entity data is saved.

## **7) Enterprise Information System Tier**

database

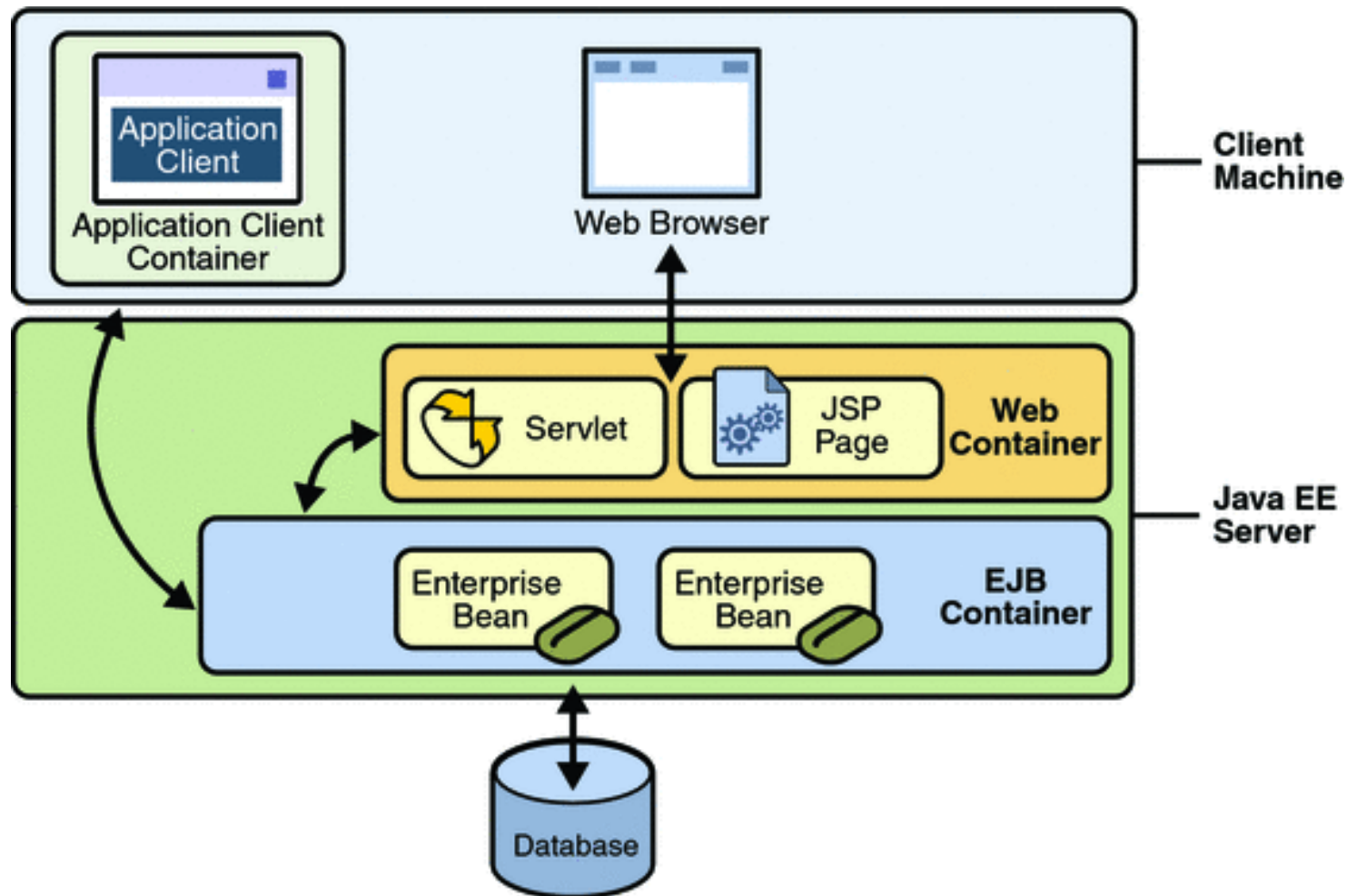
## 2.2 JavaEE Containers

Business Logic: component

Transaction、  
state management、  
multithreading、  
resource pooling, and  
other complex low-level details: ?

### Container

*Containers* are the interface between a component and the low-level platform-specific functionality that supports the component.



**Figure 2-5 JavaEE Server and Containers**

## 2.2.1 Container Services

- 1 ) a Web, enterprise bean, or application client component必须被打包成 (assembled)JavaEE module ， 该module部署在container中。
- 2 ) 打包过程(assembly process)包括每个component及JavaEE application在container中的配置：

定制 JavaEE server 的底层服务支持：

**security**： 通过配置使得只有授权的用户才能访问相关的组件。

**transaction management**： 方法中的transaction当作一个原子操作。

**Java Naming and Directory Interface (JNDI) lookups** :

访问各种命名和目录服务的统一接口。

**remote connectivity**： 访问EJB就像访问本地对象一样。

说明： Container也包含一些不可配置的services：

- (1)enterprise bean and servlet life cycles,
- (2)database connection resource pooling,
- (3)data persistence , and
- (4)access to the JavaEE platform APIs

## 2.2.2 Container Types

部署过程：安装components 于Container中。

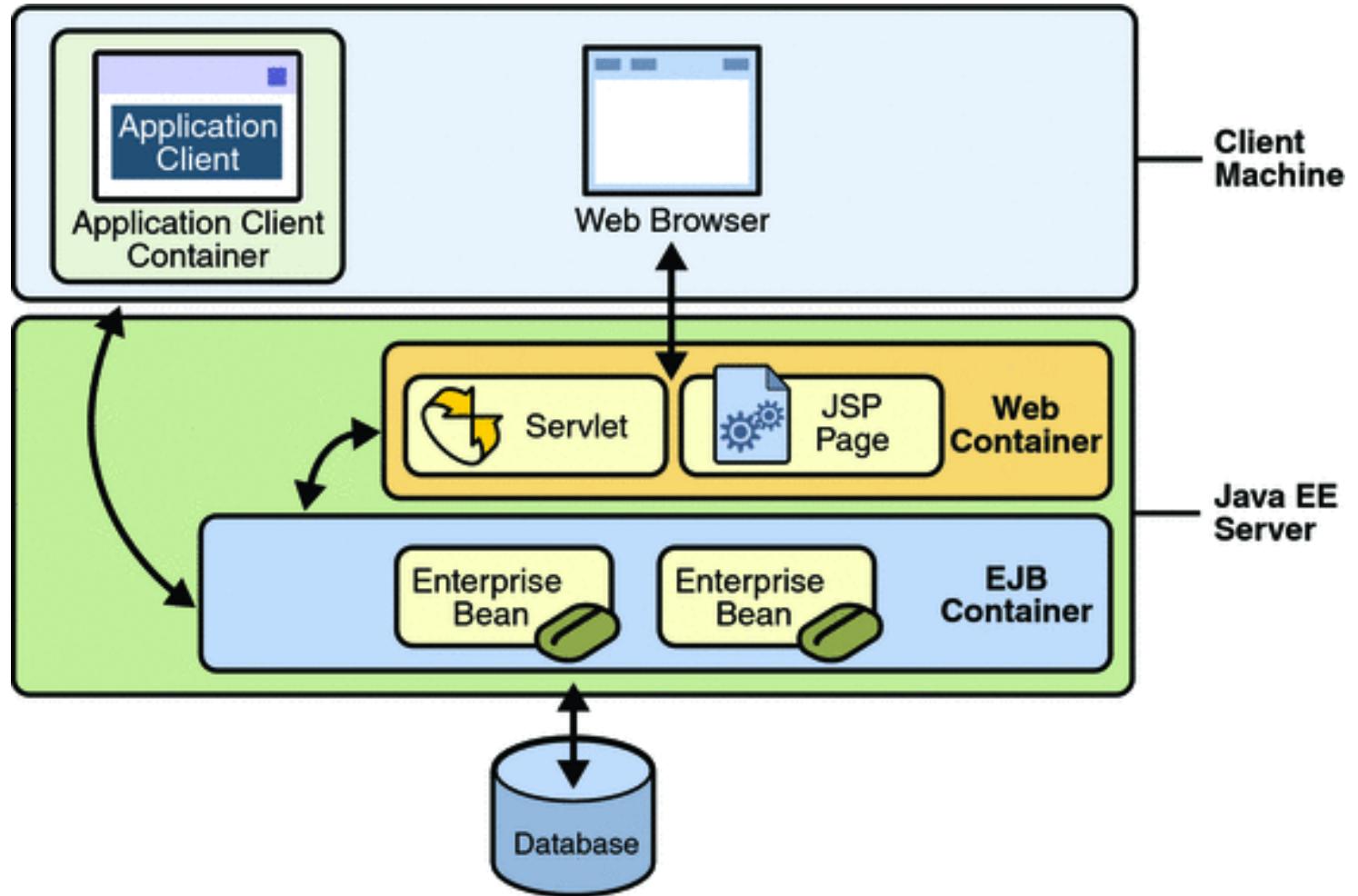


Figure 2-5 JavaEE Server and Containers

### 1 ) **JavaEE server**

The runtime portion of a JavaEE product.

A JavaEE server provides EJB and Web containers.

### 2 ) **Enterprise JavaBeans (EJB) container**

Manages the execution of enterprise beans for JavaEE applications.

Enterprise beans and their container run on the JavaEE server.

### 3 ) **Web container**

Manages the execution of JSP page and servlet components for JavaEE applications.

Web components and their container run on the JavaEE server.

### 4 ) **Application client container**

Manages the execution of application client components.

Application clients and their container run on the client.



## 2.3 Java EE Application Assembly and Deployment

- 一个**Java EE Application** (JavaEE应用系统)被打包成一个或多个标准unit,这些unit能够被部署到JavaEE服务器中。

每个 unit 包含:

- 1)一个或多个components (EJB, JSP page, servlet, applet)
- 2)一个可选的 deployment descriptor : 用来描述unit的内容。

- 部署(Deployment): 一般通过部署工具完成。

部署的目的: specify location-specific information

如: 能够访问unit的local users

local database的名字。

一旦部署完成, 应用系统就可运行。

## 2.4 Packaging Applications

### 2.4.1 Java EE application

以EAR (Enterprise Archive )文件部署在服务器中。

EAR文件： 由Java EE modules 、 deployment descriptors构成。

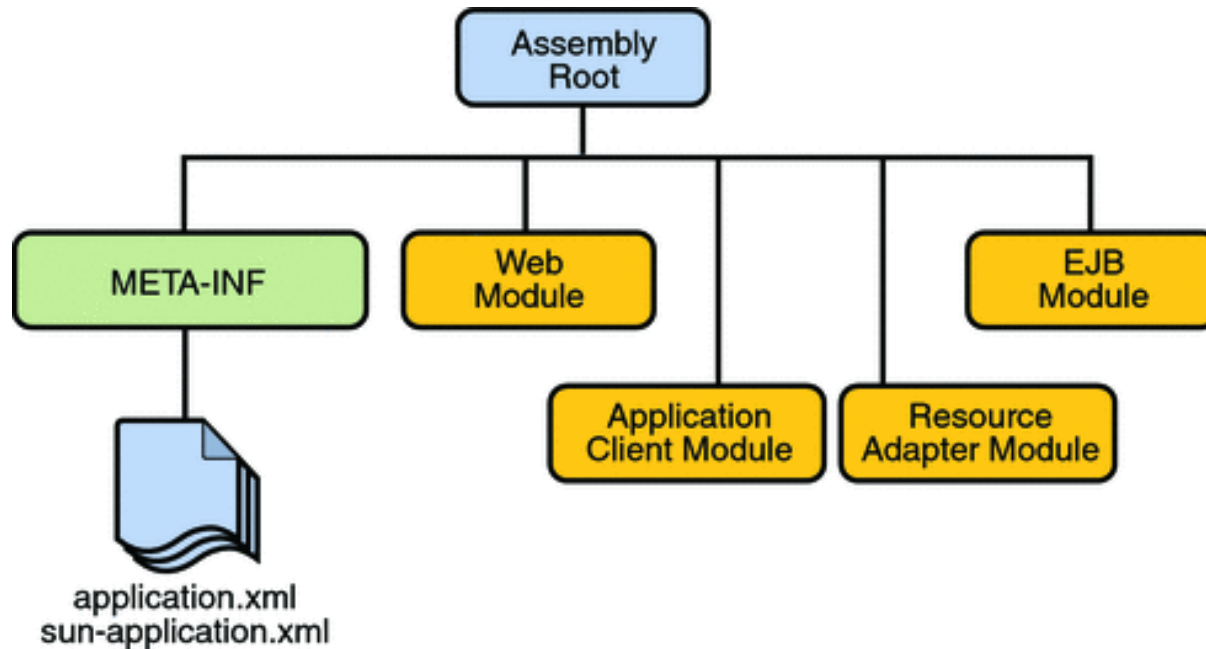


FIGURE 2-6 EAR File Structure

deployment descriptor : 描述an application, a module, or a component 的部署设置。

## 2.4.2 Java EE module :

- 1)由 一个或多个Java EE components、 一个 component deployment descriptor构成。
- 2) 可以作为一个独立的运行模块( *stand-alone* module)部署在服务器。

### 3) 类型:

EJB modules: EJB类和EJB部署文件.  
被打包为jar文件

Web modules: servlet , JSP, 公共类, GIF、HTML和  
WEB部署文件。  
被打包为war文件


Application client modules: 类和application client部署文件  
被打包为jar文件

Resource adapter modules: interfaces, classes, native  
libraries, and other documentation,  
resource adapter部署文件。  
被打包为rar文件

## 2.5 case:Library Application

### 2.5.1 应用系统目录结构及必备文件(针对JBoss4.0中间件)

[-]  LENOVO (D:)

[-]  Library

← 系统的根目录

 deploy

←

存放WEB部署文件、EJB部署文件及系统配置文件，由这些部署文件及配置文件产生应用系统的部署文件EAR

[+]  Library-ejb

←

ejb组件的根目录

[+]  Library-web

←

WEB应用程序的根目录

 META-INF

←

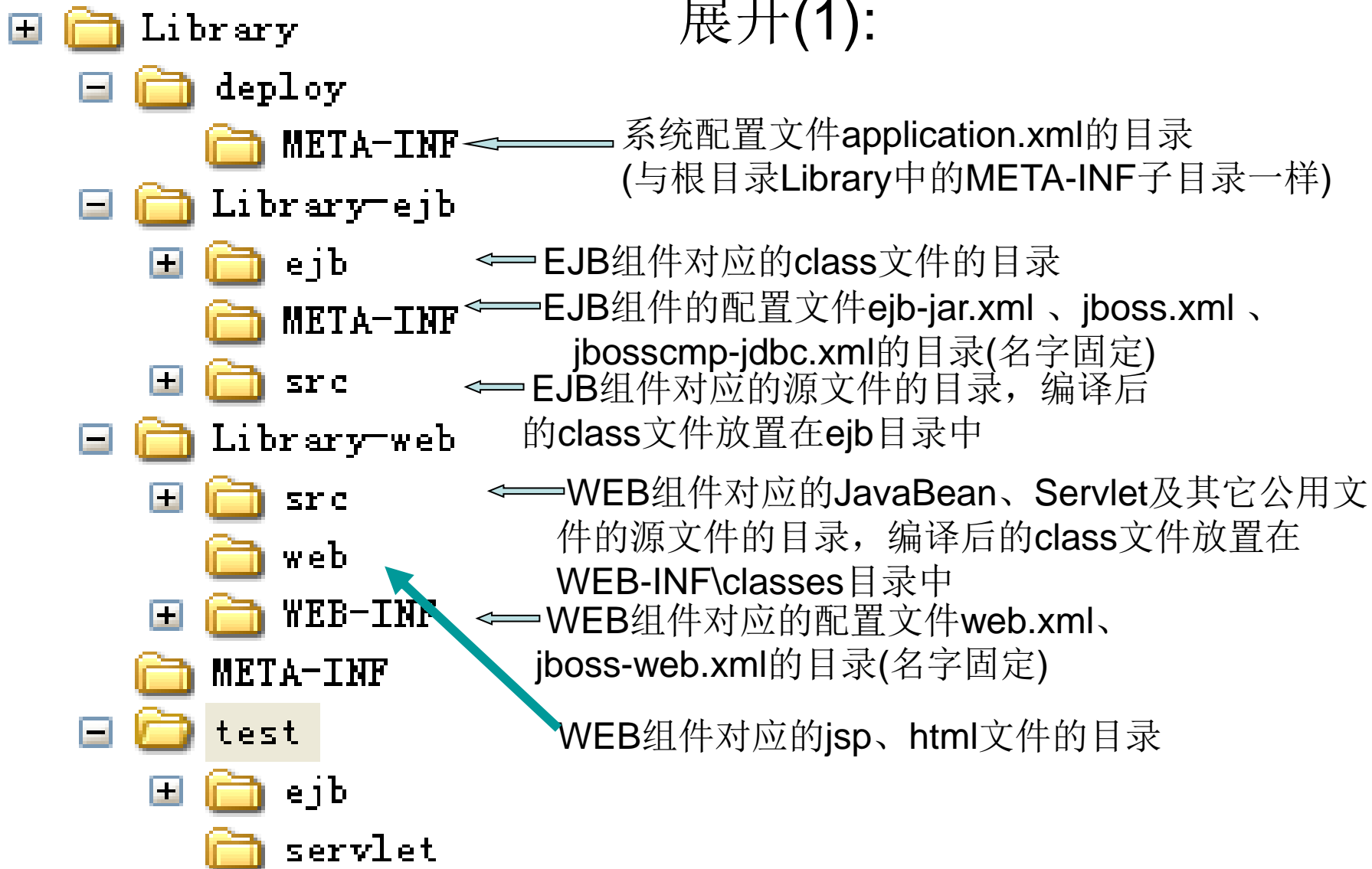
系统配置文件application.xml的目录(文件名固定)

[+]  test

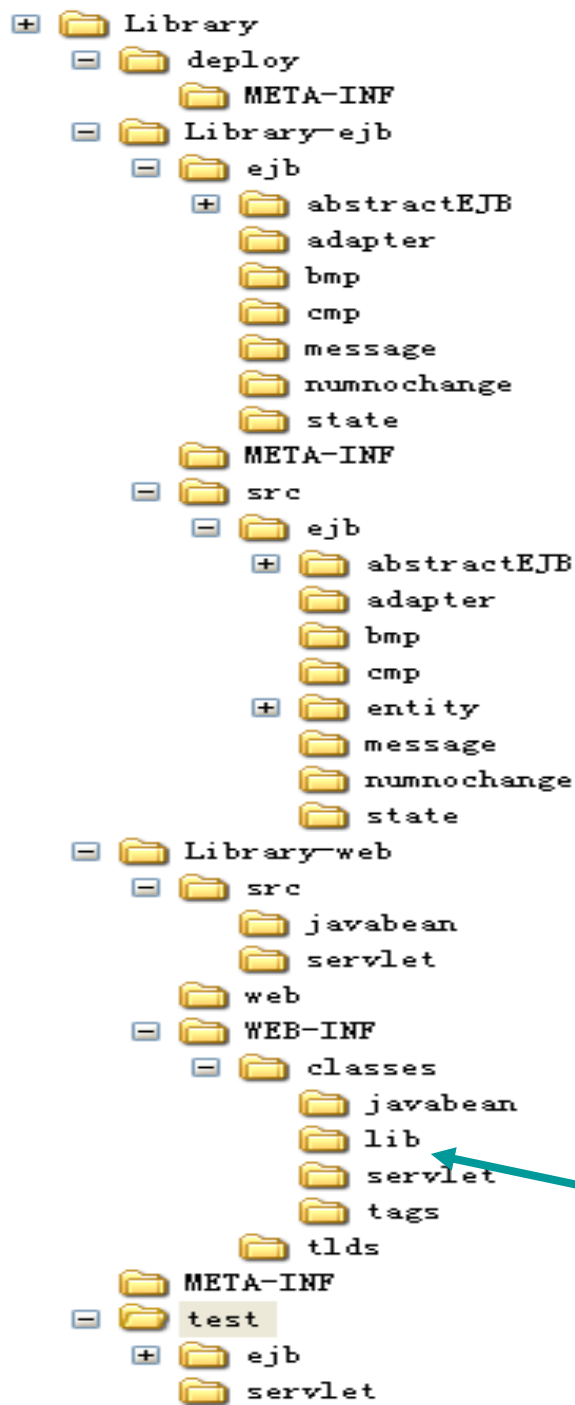
←

测试WEB组件的客户端程序(运行在命令行方式)

## 展开(1):



## 展开(2):



← EJB组件对应的class文件的目录

← EJB组件对应的源文件的目录

← WEB组件对应的JavaBean、Servlet及其它公用文件的源文件的目录，编译后的class文件放置在WEB-INF\classes目录中

← WEB组件对应的JavaBean、Servlet及其它公用文件的class文件的目录

← EJB组件、JavaBean、Servlet、JSP等java程序需要引用类对应的jar文件:mssqlserver.jar、msbase.jar、msutil.jar、javaee.jar、servlet.jar、jbossall-client.jar

# 附录: Java EE 5 APIs

