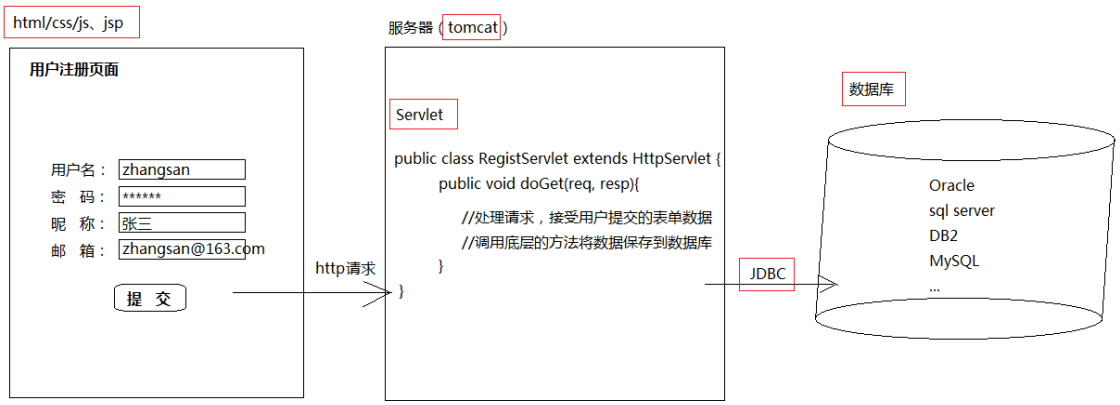
web概述

星期三,2019年7月10日　　9:20

1. web概述



a. 服务器中存储的内容：

服务器中存储web资源。

静态web资源：

HTML、css、js、jQuery

每一个用户看到的内容都相同，这些资源称之为静态web资源。

动态web资源：

JSP/Servlet

每一个用户看到的内容不相同，这些资源称之为动态web资源。

b. web即为网页，web开发即在编写完成一个网页之后，应该让Internet中的用户都可以访问。

2. 服务器概念

服务器有物理服务器和软件服务器之分。物理服务器指的是真实存储的硬件设备。软件服务器指的的一个可操作性的程序。

3. 常见服务器

WebLogic是BEA公司的产品，是目前应用最广泛的Web服务器，支持J2EE规范，而且不断的完善以适应新的开发要求，启动界面如图

另一个常用的Web服务器是IBM公司的WebSphere，支持J2EE规范，启动界面如图

在小型的应用系统或者有特殊需要的系统中，可以使用一个 免费的Web服务器：Tomcat，该服务器支持全部JSP以及Servlet规范，启动界面如图

tomcat安装和内容介绍

星期三,2019年7月10日 9:38

1. tomcat安装

解压课前资料压缩包，即可完成安装。

tomcat安装包有三种形式：

.zip

.tar.gz

.exe （安装文件）

2. tomcat启动

a. 启动方式：

在[tomcat]/bin目录下双击startup.bat文件

b. 启动时的问题：

i. tomcat窗口一闪而过：

1) 原因一：JAVA_HOME环境变量配置错误。

2) 端口号8080被占用：

a) 通过netstat -ano查看端口号及进程号

3) 安装路径中不能够出现中文和空格

c. 修改tomcat端口号：

i. [tomcat]/conf/server.xml文件中第71行，将8080修改为80.这是将tomcat服务器启动时占用的端口改为80端口。

ii. 80端口是一个缺省端口(在地址栏中可以不用书写)。

3. Tomcat的目录结构

bin --- 存放tomcat启动和关闭脚本文件的目录

conf --- 存放tomcat配置文件的目录

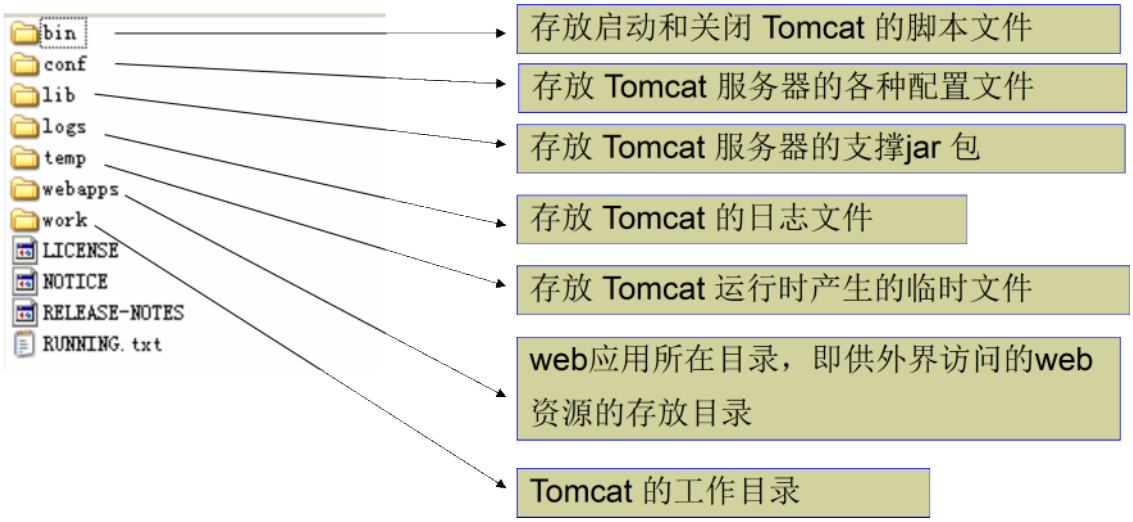
lib --- 存储tomcat在启动和运行过程中的jar包。

logs --- 存储tomcat日志信息的目录

temp --- 存储tomcat临时文件的目录

webapps --- 存储tomcat中的web应用的目录，这些应用可以被外界访问。

work --- 存储tomcat运行时产生的文件的目录

- 
bin
conf
lib
logs
temp
webapps
work
LICENSE
NOTICE
RELEASE-NOTES
RUNNING.txt
存放启动和关闭 Tomcat 的脚本文件
存放 Tomcat 服务器的各种配置文件
存放 Tomcat 服务器的支撑jar 包
存放 Tomcat 的日志文件
存放 Tomcat 运行时产生的临时文件
web应用所在目录，即供外界访问的web资源的存放目录
Tomcat 的工作目录

虚拟主机和web应用

星期三,2019年7月10日　　10:27

1. 虚拟主机概念

   在一台真实的tomcat服务器上，可以存在多个网站，这些网站在访问者看来是运行在各自独立的服务器当中，但是真实情况确实运行在同一台tomcat服务器中，所以这写网站可以称之为运行在真实服务器中的虚拟主机。一个网站就是一个虚拟主机。

   服务器也称之为主机。

2. web应用概念

   在虚拟主机中无法直接管理web资源，因为web资源涉及到的功能众多，管理维护十分繁琐。可以将同一功能的web资源放入一个目录当中，这个目录被虚拟主机管理。这样的一个目录就称之为web应用目录。一个功能创建一个web应用目录。

3. web应用部署：

   web应用是存储在当前的服务器当中，这个应用如果要被internet中的用户访问则需要配置一个虚拟路径，这个虚拟路径和真实应用的对应关系，称之为web应用的映射。

   a. web应用的映射

      i. 第一种方式：

         在conf/server.xml中添加如下内容：

         ```
         <Context path="/news1" docBase="C:/news" />
         ```

         其中path包含虚拟路径，docBase包含真实web应用的路径。

      ii. 第二种方式：

         在conf/Catalina/localhost中创建一个以虚拟路径为文件名称的.xml文件，在文件中添加<Context docBase="真实路径"/>标签。

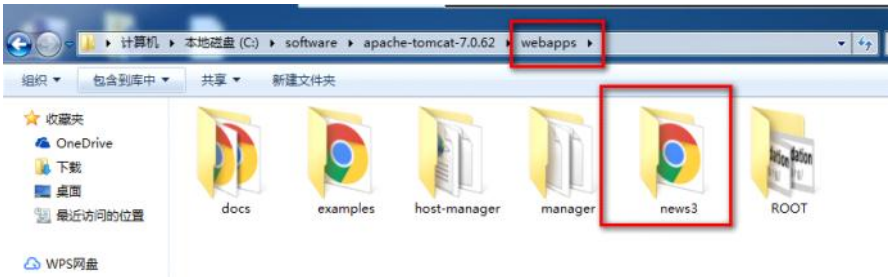
         ```
         <?xml version='1.0'  encoding='utf-8'?>
         <Context docBase="C:/news2" />
         ```

         多级目录：

         如果要求虚拟路径是多级路径，即 news2/aa,应该将xml文件的名称修改为news2#aa.xml　　在windows系统中无法使用"/"命名文件，所以使用"#"代替。

      iii. 第三种方式：

         在[tomcat]/webapps目录中添加一个新的目录，目录的名称就是web应用的虚拟路径名称。这个目录就是真实的web应用。

b. 缺省web应用的配置：

所谓缺省web应用，就是配置一个web应用的虚拟路径为缺省路径(可以不用书写)。

i. 第一种方式：

将<Context path="" docBase="真实路径"/>中的path属性值设置为""即可。

ii. 第二种方式：

将conf/Catalina/localhost中的.xml文件名称改为ROOT.xml文件，这个ROOT即为缺省路径的名称。

iii. 第三种方式：

将webapps中的web应用名称改为ROOT，即可作为缺省的web应用使用。

iv. 路径缺省的优先级：

server.xml >Catalina/localhost/ROOT.xml > webapps/ROOT

4. web应用目录结构

一个web应用的目录结构可以只包含一些静态资源，但是同样可以创建一个完成的web资源（可以包含动态资源）.完整的目录结构如下：

news
|
|--- 1.html web应用目录中可以存放静态资源，这些资源可以被浏览器直接访问。
|
|---WEB-INF 一个web应用如果有这个目录，则一定要保证这个目录结构的完整，否则可能会导致web应用无法工作。
|静态web资源 此处的静态资源无法被浏览器直接访问。
|---lib 存放web应用所依赖的jar包。
|---classes 存放编译后的class文件的目录。(动态资源存放的目录)
|---web.xml 用于配置当前web应用的过滤器，监听器，servlet映射，缺省主页的文件。

5. web.xml文件

一个完整的web应用需要包含WEB-INF目录，其中必然会有一个web.xml文件。这个文件是当前web应用的配置文件，可以在其中书写过滤器，监听器，servlet映射，缺省主页等配置信息。

i. web.xml功能：

配置过滤器
配置监听器
配置servlet映射
配置缺省主页。全站首页。(conf/web.xml配置缺省首页)

ii. web.xml应用：配置缺省主页：

从conf/web.xml中复制声明及起始标签，和最后的welcome-file-list标签及结束标签。这样基本的web.xml文件结构就已经形成。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
 version="3.0">


   <welcome-file-list>
     <welcome-file>1.html</welcome-file>
   </welcome-file-list>

</web-app>
```

a. 注意：所有的web应用都会继承tomcat/conf/web.xml文件，其中的配置信息在每一个web应用中都可以使用。

6. 虚拟主机的配置
   a. 修改server.xml：

修改其中的<Engine>标签的<Host>标签，添加内容如下：

```
<Host name="www.baidu.com"  appBase="baidu">


   </Host>
```

name属性为当前虚拟主机的名称(也是域名)

appBase属性为当前虚拟主机管理的目录。(目录的路径可以是相对会绝对路径。)

   b. 修改hosts文件

i. 在配置host标签之后，仍然无法访问这个网站。根本原因是www.baidu.com没有指向本地，可以修改hosts文件让网站指向本地。

1）修改文件的路径：

```
C:\Windows\System32\drivers\etc\hosts
```

2）修改hosts文件的原因：

hosts文件可以将域名解析为ip地址.(在访问一个网站的时候最后会根据ip地址访问。)

- ~<Host>有这些属性：

属性	描述	必需？
className	默认为org.apache.catalina.core.StandardHost	否
appBase	web应用程序文件存放的位置，相对路径为CATALINA_HOMEM	是
autoDeploy	设为true，则web.xml发生变化时，tomcat自动重新部署程序。实现这个功能必需允许后台处理	否
name	virtual host的名称	是
backgroundProcessingDelay	跟Engine中的backgroundProcessingDelay类似	否
deployOnStartup	若为true，则当这个Engine启动时，tomcat将自动部署这个host，默认为true	否

deployXML	这个属性的目的是为了提高tomcat的安全性，控制web应用程序是否能使用META-INF/contex.xml。如果设为false，则各应用程序只能访问$CATALINA_HOME/conf/<engine>/<host>/<app>.xml。默认值为True。	否
errorReportValveClass	定义host使用的error-reporting Valve，默认值为org.apache.catalina.valves.ErrorReportValve	否
unpackWARs	tomcat在webapps文件夹中发现war文件时，是否自动将其解压	否
workdir	tomcat使用这个目录来放工作着的servlet和jsp（以servlet形式），这里面的servlet都是是编译好的class文件。默认为$CATALINA_HOME/work	

c. 缺省虚拟主机

在用户使用一个ip地址访问对应的服务器时，服务器无法确定使用哪一台虚拟主机提供服务器，这时会采用缺省虚拟主机提供服务。

i. 缺省虚拟主机的配置：

在<Engine>标签上有一个defaultHost属性，其中包含的内容就是缺省虚拟主机，在使用ip访问的时候，这个虚拟主机就会提供服务。配置如下：

```
<Engine name="Catalina" defaultHost="www.baidu.com">
```

7. 杂项

a. 打war包。

i. 在某一个web应用的目录下访问cmd窗口，在窗口中书写：

> jar -cvf ROOT.war *

ii. 进入到某一个web应用的目录中，将所有文件添加到一个.zip的压缩包中，并修改后缀为.war

b. ~web.xml 和Context.xml文件

Context.xml中指定的了每一个web应用监听的资源路径 ,路径内容为WEB-INF/web.xml文件.

conf目录下的web.xml被所有的web引用的web.xml文件继承。

8. 注意：

在server.xml中如果配置了缺省web应用，则运用第三种发布方式自动部署一个ROOT.war包是不可以实现的。需要将第一种web应用的缺省路径修改为其他路径内容。这样第三种方式就可以自动发布一个缺省web应用。

9. 作业：

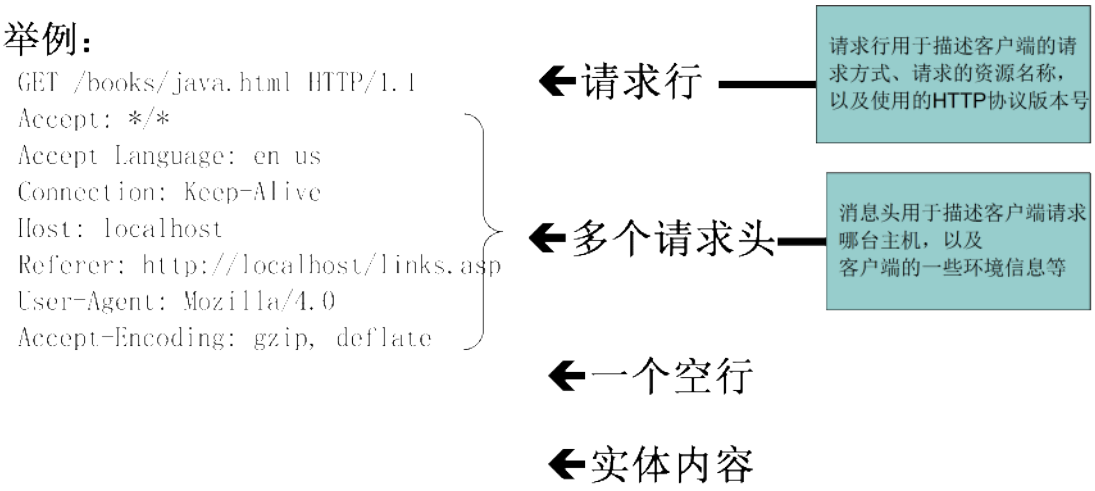
构建一个www.google.com，并且作为缺省虚拟主机使用。其中包含news应用和mail两个。将mail修改为缺省web应用，并配置缺省主页。

HTTP协议

星期三,2019年7月10日　　16:14

1. HTTP协议概述
   a. HTTP　HyperText　transfor protocol(超文本传输协议)。
   b. HTTP协议就是用来规定发送数据的格式的一个标准。HTTP在传输数据是将数据转换为一定格式，HTTP协议指定了数据传输的格式。
2. HTTP协议模型：
   基于请求响应模型。
   一次请求对应一次响应。
   请求由浏览器发出，服务器根据请求作出响应。
3. HTTP协议版本
   a. HTTP/1.0
      在浏览器和服务器建立连接之后，浏览器发出一次请求，服务器作出一次响应，连接就会断开。
   b. HTTP/1.1
      在浏览器和服务器建立连接之后，浏览器可以依次发出多次请求，服务器会根据各个请求依次作出响应。在服务器等待一段时间之后，如果没有更多的请求到达，则会将连接断开。
4. ~~~Telnet指定模拟HTTP协议工作
   - 利用telnet演示HTTP1.0和HTTP1.1的区别
     - 首先准备一个web应用
     - 打开cmd窗口，输入telnet localhost 端口号
     - 按crtl+]
     - 回车，进入输入界面（输入界面不允许删除字符，因为每输入一个字符就将这个字符以流的形式发送给服务器，无法修改。）
     - 输入指令　GET　/web应用/资源名　HTTP/1.1
     - 回车，输入Host:localhost:端口号
   - 使用HTTP1.1协议和1.0协议分别测试，发现1.1执行完一次请求后等待下一次请求，1.0则在一次请求后断开。
   - telnet测试注意事项：每次telnet测试完成后要退出重进，必要时需要重启tomcat。
   - 一个好多同学搞不清楚的问题：
     - 一个web页面中，使用img标签引用了三幅图片，当浏览器访问服务器中的这个web页面时，浏览器总共会请求几次服务器？
       总共会发送4次请求。只会建立一次连接
5. HTTP请求
   - 客户端连上服务器后，向服务器请求某个web资源，称之为客户端向服务器发送了一个HTTP请求。一个完整的HTTP请求包括如下内容：

- 一个请求行、若干请求头、一个空行、以及实体内容，如下所示 ：
  - 举例：



6. HTTP请求的细节——请求行

- 请求行中的GET称之为请求方式，请求方式有：
- POST、 GET、 HEAD、 OPTIONS、 DELETE、 TRACE、 PUT
- 常用的有： GET、 POST
- 用户如没有设置，默认情况下浏览器向服务器发送的都是get请求，例如在浏览器直接输地址访问，点超链接访问等都是get，用户如想把请求方式改为post，可通过更改表单的提交方式实现。

- 不管POST或GET，都用于向服务器请求某个WEB资源，这两种方式的区别主要表现在数据传递上：
  - 如请求方式为GET方式，则可以在请求的URL地址后以?的形式带上交给服务器的数据，多个数据之间以&进行分隔，例如：
    - GET /mail/1.html?name=abc&password=xyz HTTP/1.1
  - GET方式的特点：在URL地址后附带的参数是有限制的，其数据容量通常不能超过1K。
  - 如请求方式为POST方式，则可以在请求的实体内容中向服务器发送数据，Post方式的特点：传送的数据量无限制。

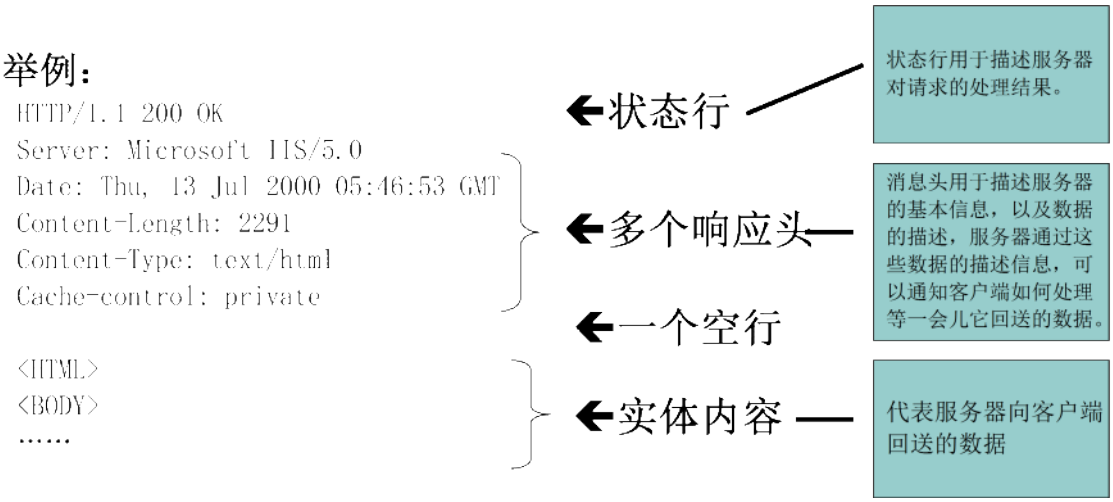
7. HTTP请求的细节——请求头

- 用于HTTP请求中的常用头 。
- Accept: text/html,image/*　通知浏览器可以接受什么数据 现在表示接收文本数据,任意格式的图片数据
- Accept-Charset: ISO-8859-1
- Accept-Encoding: gzip,compress　发送数据的压缩格式
- Accept-Language: en-us,zh-cn　语言
- Host: www.it315.org:80　请求的地址,可以通过观察Host头来确定访问对应地址服务器上的哪个虚拟主机。
- If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT 与响应头last modified呼应
- Referer: http://www.it315.org/index.jsp　防盗链
- User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
- Cookie

- Connection: close/Keep-Alive
- Date: Tue, 11 Jul 2000 18:23:51 GMT

8. HTTP响应
   - 一个HTTP响应代表服务器向客户端回送的数据，它包括：
   - 一个状态行、若干响应头(消息头)、一个空行、以及实体内容 。

   - 举例：



9. HTTP响应的细节——状态行

状态行

格式： HTTP版本号　状态码　原因叙述<CRLF>

举例：HTTP/1.1 200 OK

状态码用于表示服务器对请求的处理结果，它是一个三位的十进制数。响应状态码分为5类，如下所示：

状态码	含义
100～199	表示成功接收请求，要求客户端继续提交下一次请求才能完成整个处理过程
200～299	表示成功接收请求并已完成整个处理过程，常用200
300～399	为完成请求，客户需进一步细化请求。例如，请求的资源已经移动一个新地址，常用302、307和304
400～499	客户端的请求有错误，常用404
500～599	服务器端出现错误，常用 500

302 +location　304　307---缓存

404　访问资源路径不存在

500 505　503　502　服务器报错(代码编写出现错误)

10. HTTP响应细节——常用响应头
    - HTTP请求中的常用响应头
    - Location: http://www.it315.org/index.jsp　配合302实现请求重定向
    - Server:apache tomcat　服务器类型
    - Content-Encoding: gzip　服务器发送数据的压缩格式
    - Content-Length: 80　发送数据的长度
    - Content-Language: zh-cn　发送数据的语言环境

- Content-Type: text/html; charset=GB2312　可接受数据格式和语言
- Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 与请求头的if modified since头呼应
- Refresh: 1;url=http://www.it315.org　定时跳转
- Content-Disposition: attachment;filename=aaa.zip
- Transfer-Encoding: chunked
- Set-Cookie:SS=Q0=5Lb_nQ; path=/search
- ETag: W/"83794-1208174400000"
- Expires: -1 通知浏览器是否缓存当前资源，如果这个头的值是一个以毫秒为单位的值就是通知浏览器缓存资源到指定的时间点，如果值是0或-1则是通知浏览器禁止缓存。
- Cache-Control: no-cache 通知浏览器是否缓存的头
- Pragma: no-cache 通知浏览器是否缓存的头
- Connection: close/Keep-Alive
- Date: Tue, 11 Jul 2000 18:23:51 GMT

11. ~OSI网络七层协议

    应用层（HTTP、FTP、SMTP、POP3、TELNET）->表示层->会话层->传输层（TCP、UDP）->网络层（IP）->数据链路层->物理层

    TCP/IP四层模型

JavaEE规范

星期三,2019年7月10日　　9:33

java的大方向就是JavaEE，JavaEE不仅仅是socket编程，具体包括13中核心技术。

JAVAEE的核心API与组件

JAVAEE平台由一整套服务（Services）、应用程序接口（APIs）和协议构成，它对开发基于Web的多层应用提供了功能支持，下面对JAVAEE中的13种技术规范进行简单的描述(限于篇幅，这里只进行简单的描述)：

1、JDBC(Java Database Connectivity)　　JDBC API为访问不同的数据库提供了一种统一的途径，象ODBC一样，JDBC对开发者屏蔽了一些细节问题，另外，JDCB对数据库的访问也具有平台无关性。

2、JNDI(Java Name and Directory Interface)　　JNDI API被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源如DNS和LDAP，本地文件系统，或应用服务器中的对象。

3、EJB(Enterprise JavaBean)　　JAVAEE技术之所以赢得媒体广泛重视的原因之一就是EJB。它们提供了一个框架来开发和实施分布式商务逻辑，由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB规范定义了EJB组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB并不是实现JAVAEE的唯一途径。正是由于JAVAEE的开放性，使得有的厂商能够以一种和EJB平行的方式来达到同样的目的。

4、RMI(Remote Method Invoke)　　正如其名字所表示的那样，RMI协议调用远程对象上方法。它使用了序列化方式在客户端和服务器端传递数据。RMI是一种被EJB使用的更底层的协议。

5、Java IDL/CORBA　　在Java IDL的支持下，开发人员可以将Java和CORBA集成在一起。他们可以创建Java对象并使之可在CORBA ORB中展开, 或者他们还可以创建Java类并作为和其它ORB一起展开的CORBA对象的客户。后一种方法提供了另外一种途径，通过它Java可以被用于将你的新的应用和旧的系统相集成。

6、JSP(Java Server Pages)　　JSP页面由HTML代码和嵌入其中的Java代码所组成。服务器在页面被客户端所请求以后对这些Java代码进行处理，然后将生成的HTML页面返回给客户端的浏览器。

7、Java Servlet　　Servlet是一种小型的Java程序，它扩展了Web服务器的功能。作为一种服务器端的应用，当被请求时开始执行，这和CGI Perl脚本很相似。Servlet提供的功能大多与JSP类似，不过实现的方式不同。JSP通常是大多数HTML代码中嵌入少量的Java代码，而servlets全部由Java写成并且生成HTML。

8、XML(Extensible Markup Language)　　XML是一种可以用来定义其它标记语言的语言。它被用来在不同的商务过程中共享数据。 XML的发展和Java是相互独立的，但是，它和Java具有的相同目标正是平台独立性。通过将Java和XML的组合，您可以得到一个完美的具有平台独立性的解决方案。

9、JMS(Java Message Service)　　JMS是用于和面向消息的中间件相互通信的应用程序接口(API)。它既支持点对点的域，有支持发布/订阅(publish/subscribe)类型的域，并且提供对下列类型的支持：经认可的消息传递,事务型消息的传递，一致性消息和具有持久性的订阅者支持。JMS还提供了另 一种方式来对您的应用与旧的后台系统相集成。

10、JTA(Java Transaction Architecture)　　JTA定义了一种标准的API，应用系统由此可以访问各种事务监控。

11、JTS(Java Transaction Service)　　JTS是CORBA OTS事务监控的基本的实现。JTS规定了事务管理器的实现方式。该事务管理器是在高层支持Java Transaction API (JTA)规范，并且在较底层实现OMG OTS specification的Java映像。JTS事务管理器为应用服务器、资源管理器、独立的应用以及通信资源管理器提供了事务服务。

12、JavaMail　　JavaMail是用于存取邮件服务器的API，它提供了一套邮件服务器的抽象类。不仅支持SMTP服务器，也支持IMAP服务器。

13、JAF(JavaBeans Activation Framework)　　JavaMail利用JAF来处理MIME编码的邮件附件。MIME的字节流可以被转换成Java对象，或者转换自Java对象。大多数应用都可以不需要直接使用JAF