Servlet

# Servlet的基本概念

|  |
| --- |
| A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.  servlet是一个java小程序运行在web服务器。servlet接收并响应来自Web客户端的请求，通常通过HTTP，超文本传输协议。 |

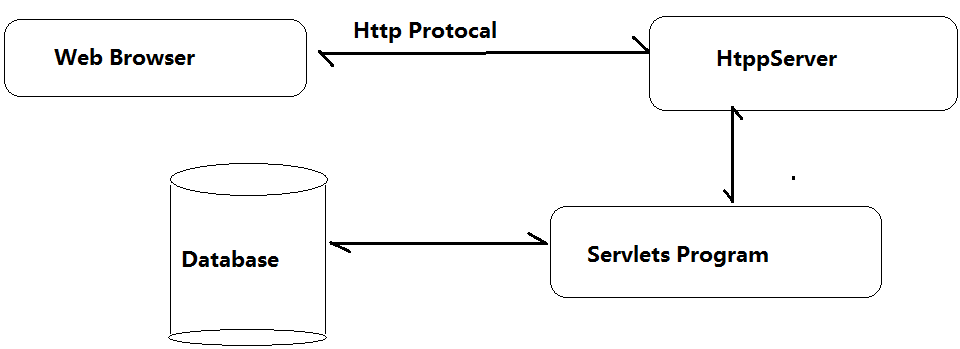
## 主要作用：

|  |
| --- |
| * 接收浏览器（客户端）的请求数据；（接收由表单用户输入的数据） * 给客户端做响应； * 呈现来自数据库或者其他源的记录； * 动态创建网页 |

*接收页面的数据------->接收数据------>-处理数据------>-给客户端做出相应的响应数据*

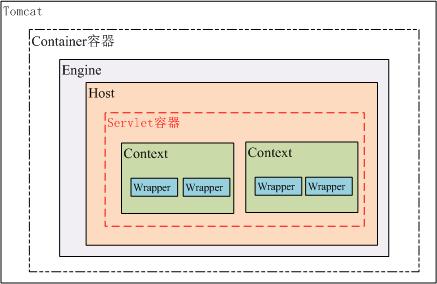
## Servlet 在web应用程序的位置

**Java Servlet 是运行在 Web 服务器或应用服务器上的程序，它是作为来自 Web 浏览器请求和 HTTP 服务器上的数据库中间层。**



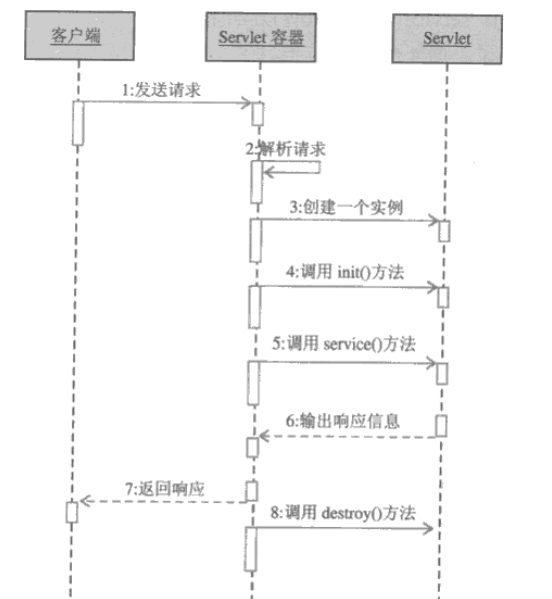
## Servlet 运行在Tomcat容器

Tomcat 的容器等级中，Context 容器是直接管理 Servlet 在容器中的包装类 Wrapper，所以 Context 容器直接影响 Servlet 的工作方式



从上图可以看出 Tomcat 的容器分为四个等级，真正管理 Servlet 的容器是 Context 容器，一个 Context 对应一个 Web 工程。Context容器中会有多个servlet。

## Servlet的生命周期



我们创建的servlet服务器（Tomcat）帮助我们完成解析工作，并且被包装成 StandardWrapper 添加在 Context 容器中，通过实例化以后就能为我们工作。如何实例化？

* 用户在浏览器中输入一个 URL。Web 服务器配置文件确定该 URL 是否指向一个由运行于服务器上的 servlet 容器所管理的 servlet。
* 如果还没有创建该 servlet 的一个实例（一个应用程序只有一个 servlet 实例），那么该容器就加载该类，并将之实例化。
* 该容器调用 servlet 上的 init()。
* 该容器调用servlet 上的service()，并在包装的 HttpServletRequest 和 HttpServletResponse 中进行传递。
* 该 servlet 通常访问请求中的元素，代表其他服务器端类来执行所请求的服务并访问诸如数据库之类的资源，然后使用该信息填充响应。
* 如果有必要，在 servlet 的有用生命结束时，该容器会调用 servlet 上的 destroy() 来清除它。

### 实例化

当客户端第一次进行请求的时候会实例化具体的servlet

如果 Servlet 的 load-on-startup 配置项大于 0，那么在 Context 容器启动的时候就会被实例化，在web.xml 中配置，也就是当 Tomcat 启动时这个 Servlet 就会被启动。

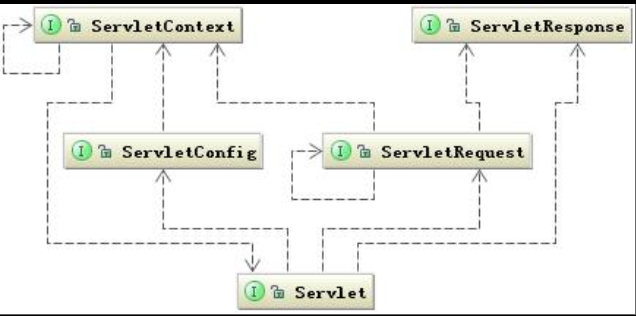
Servlet的加载顺序：当在web.xml中配置标签 <load-on-startup>2</load-on-startup>其中的数值大于0的整数时，当服务器启动的时候会根据配置的数值的大小顺序进行加载实例

化servlet大于对象并执行初始化方法 。

1-->2-->3 -------加载顺序-----1-----2----3

（因为Tomcat内部配置了一个1，所以我们的配置一般从2开始）

## Servlet的体系结构



与 Servlet 主动关联的是三个类，分别是 ServletConfig、ServletRequest 和 ServletResponse。这三个类都是通过容器传递给 Servlet 的。

创建servlet以后，服务器帮助我们将servlet加载到容器中，当客户端请求到服务器的时候，Tomcat接收请求的时候会解析用户的请求由哪个servlet来服务----->第一次实例化这个servlet。其中 ServletConfig 是在 Servlet 初始化时就传给 Servlet 了，而后两个是在请求达到时调用 Servlet 时传递过来的。

# Servlet中的常用对象

## ServletConfig

### ServletConfig对象的获取

这个对象在 Servlet 初始化时就传给 Servlet ，所以从servlet对象中去取。

this.getServletConfig ();

### ServletConfig对象的作用

* 获取当前servlet的初始化参数

**什么是初始化参数**

动servlet的时候给他初始化一些参数,这样我们可以在web.xml配置文件中对servlet进行配置。.当servlet配置了初始化参数后，web容器在创建servlet实例对象时， 会自动将这些初始化参数封装到ServletConfig对象中，并在调用servlet的init方法时，将ServletConfig对象传递给servlet。进而，程序员通过ServletConfig对象就可以得到当前servlet的初始化参数信息。

### 常用方法

|  |  |
| --- | --- |
| java.lang.String | **[getInitParameter](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletConfig.html" \l "getInitParameter(java.lang.String))**(java.lang.String name)            Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist. |
| java.util.Enumeration | **[getInitParameterNames](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletConfig.html" \l "getInitParameterNames())**()            Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters. |
| [ServletContext](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html) | **[getServletContext](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletConfig.html" \l "getServletContext())**()            Returns a reference to the [ServletContext](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html) in which the caller is executing. |
| java.lang.String | **[getServletName](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletConfig.html" \l "getServletName())**()            Returns the name of this servlet instance. |

## ServletContext

servletContext对象是Servlet三大域对象之一，每个Web应用程序都拥有一个ServletContext对象，该对象是Web应用程序的全局对象或者上下文。Tomcat服务器在启动时，会自动创建一个ServletContext对象，域产生 。在关闭时，会自动销毁这个ServletContext对象。每个Web应用程序只拥有一个ServletContext对象，ServletContext对象可以在整个Web应用中共享数据资源。

### 获取ServletContext对象的方法

this.getServletConfig.getServletContext();

This.getServletContex() ;

### ServletContext对象的作用

* **获取全局的初始化参数**

context.getInitParameter(String name);context.getInitParameterNames() ;

* **实现整个web应用程序中多个servlet之间的数据共享(application域)**

context.setAttributer()； context.getAttributer()

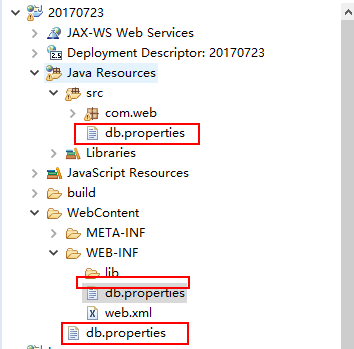
* **实现转发:一般使用request对象来转发**
* **获取文件的真实路径和项目名路径（项目名）**

getRealPath（）；方法来获取web项目中文件的真实路径

第一种 src目录下的文件 /WEB-INF/classes/db.properties(src目录下的东西都发布在/WEB-INF/classes/目录下面)

第二种WebContent目录下的文件/db.properties(默认在跟目录下)

第三种 WEB-INF目录下的文件/WEB-INF/db.properties



·········

getContextPath() 获得路径--->/20170723(也就是项目名)

常用方法

|  |  |
| --- | --- |
| java.lang.Object | **[getAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getAttribute(java.lang.String))**(java.lang.String name)            Returns the servlet container attribute with the given name, or null if there is no attribute by that name. |
| java.util.Enumeration | **[getAttributeNames](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getAttributeNames())**()            Returns an Enumeration containing the attribute names available within this servlet context. |
| java.lang.String | **[getInitParameter](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getInitParameter(java.lang.String))**(java.lang.String name)            Returns a String containing the value of the named context-wide initialization parameter, or null if the parameter does not exist. |
| java.util.Enumeration | **[getInitParameterNames](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getInitParameterNames())**()            Returns the names of the context's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the context has no initialization parameters. |
| java.lang.String | **[getRealPath](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getRealPath(java.lang.String))**(java.lang.String path)            Returns a String containing the real path for a given virtual path. |
| [RequestDispatcher](mk:@MSITStore:D:/jsdk23/javax/servlet/RequestDispatcher.html) | **[getRequestDispatcher](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getRequestDispatcher(java.lang.String))**(java.lang.String path)            Returns a [RequestDispatcher](mk:@MSITStore:D:/jsdk23/javax/servlet/RequestDispatcher.html) object that acts as a wrapper for the resource located at the given path. |
| java.net.URL | **[getResource](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getResource(java.lang.String))**(java.lang.String path)            Returns a URL to the resource that is mapped to a specified path. |
| java.io.InputStream | **[getResourceAsStream](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "getResourceAsStream(java.lang.String))**(java.lang.String path)            Returns the resource located at the named path as an InputStream object. |
| void | **[removeAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "removeAttribute(java.lang.String))**(java.lang.String name)            Removes the attribute with the given name from the servlet context. |
| void | **[setAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletContext.html" \l "setAttribute(java.lang.String, java.lang.Object))**(java.lang.String name, java.lang.Object object)            Binds an object to a given attribute name in this servlet context. |

## ServletRequest

当servlet被调用的时候会传递过来request对象，这个对象用来处理请求。

主要作用：

**1 接收页面传递过来的请求参数**

**2 作为一个请求域对象存储数据**

**3 实现请求的转发**

**4 可以获取当前用户的session对象和cookie对象**

**5 获取当前请求的请求路径（URL）**

**6 可以获取项目路径**

ServletRequest对象的常用方法

|  |  |
| --- | --- |
| java.lang.Object | **[getAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getAttribute(java.lang.String))**(java.lang.String name)            Returns the value of the named attribute as an Object, or null if no attribute of the given name exists. |
| java.util.Enumeration | **[getAttributeNames](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getAttributeNames())**()            Returns an Enumeration containing the names of the attributes available to this request. |
| java.lang.String | **[getCharacterEncoding](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getCharacterEncoding())**()            Returns the name of the character encoding used in the body of this request. |
| java.lang.String | **[getParameter](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getParameter(java.lang.String))**(java.lang.String name)            Returns the value of a request parameter as a String, or null if the parameter does not exist. |
| java.util.Map | **[getParameterMap](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getParameterMap())**()            Returns a java.util.Map of the parameters of this request. |
| java.util.Enumeration | **[getParameterNames](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getParameterNames())**()            Returns an Enumeration of String objects containing the names of the parameters contained in this request. |
| java.lang.String[] | **[getParameterValues](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "getParameterValues(java.lang.String))**(java.lang.String name)            Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist. |
| void | **[removeAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "removeAttribute(java.lang.String))**(java.lang.String name)            Removes an attribute from this request. |
| void | **[setAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "setAttribute(java.lang.String, java.lang.Object))**(java.lang.String name, java.lang.Object o)            Stores an attribute in this request. |
| void | **[setCharacterEncoding](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletRequest.html" \l "setCharacterEncoding(java.lang.String))**(java.lang.String env)            Overrides the name of the character encoding used in the body of this request. |

HttpServletRequest常用方法

|  |  |
| --- | --- |
| String | **[getContextPath](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getContextPath())**()            Returns the portion of the request URI that indicates the context of the request. |
| [Cookie](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html)[] | **[getCookies](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getCookies())**()            Returns an array containing all of the Cookie objects the client sent with this request. |
| java.lang.String | **[getHeader](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getHeader(java.lang.String))**(java.lang.String name)            Returns the value of the specified request header as a String. |
| java.util.Enumeration | **[getHeaderNames](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getHeaderNames())**()            Returns an enumeration of all the header names this request contains. |
| java.util.Enumeration | **[getHeaders](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getHeaders(java.lang.String))**(java.lang.String name)            Returns all the values of the specified request header as an Enumeration of String objects. |
| java.lang.String | **[getMethod](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getMethod())**()            Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT |
| java.lang.String | **[getRequestURI](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getRequestURI())**()            Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request. |
| java.lang.StringBuffer | **[getRequestURL](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getRequestURL())**()            Reconstructs the URL the client used to make the request. |
| java.lang.String | **[getServletPath](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getServletPath())**()            Returns the part of this request's URL that calls the servlet. |
| [HttpSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html) | **[getSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession())**()            Returns the current session associated with this request, or if the request does not have a session, creates one. |
| [HttpSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html) | **[getSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession(boolean))**(boolean create)            Returns the current HttpSession associated with this request or, if if there is no current session and create is true, returns a new session. |

## ServletResponse

和request对象一样在servlet创建的时候这个对象就会被放在service方法的参数中 ，

主要作用：

1对客户端（浏览器）进行响应 --->响应字符流，响应字节流

2返回客户端（浏览器）cookie对象

3实现重定向

ServletResponse常用方法

|  |  |
| --- | --- |
| java.lang.String | **[getCharacterEncoding](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletResponse.html" \l "getCharacterEncoding())**()            Returns the name of the charset used for the MIME body sent in this response. |
| [ServletOutputStream](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletOutputStream.html) | **[getOutputStream](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletResponse.html" \l "getOutputStream())**()            Returns a [ServletOutputStream](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletOutputStream.html) suitable for writing binary data in the response. |
| java.io.PrintWriter | **[getWriter](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletResponse.html" \l "getWriter())**()            Returns a PrintWriter object that can send character text to the client. |
| void | **[setContentType](mk:@MSITStore:D:/jsdk23/javax/servlet/ServletResponse.html" \l "setContentType(java.lang.String))**(java.lang.String type)            Sets the content type of the response being sent to the client. |

  HttpServletResponse常用方法

|  |  |
| --- | --- |
| void | **[sendRedirect](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletResponse.html" \l "sendRedirect(java.lang.String))**(java.lang.String location)            Sends a temporary redirect response to the client using the specified redirect location URL. |
| void | **[addCookie](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletResponse.html" \l "addCookie(javax.servlet.http.Cookie))**([Cookie](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html) cookie)            Adds the specified cookie to the response. |

## Session 与 Cookie

**会话技术----什么是会话**

简单来说就是打开浏览器，在一个站点中点击各种超链接，直到关闭这个网站所有的页面，或者关闭浏览器，这个过程就叫做一次会话 。

**会话技术保存用户会话时产生的数据信息**

Session 与 Cookie 的作用都是为了保持访问用户与后端服务器的交互状态。它们有各自的优点也有各自的缺陷。然而具有讽刺意味的是它们优点和它们的使用场景又是矛盾的，例如使用 Cookie 来传递信息时，随着 Cookie 个数的增多和访问量的增加，它占用的网络带宽也很大，试想假如 Cookie 占用 200 个字节，如果一天的 PV 有几亿的时候，它要占用多少带宽。所以大访问量的时候希望用 Session，但是 Session 的致命弱点是不容易在多台服务器之间共享，所以这也限制了 Session 的使用。

### Session和cookie本身存在的问题：

1 使用cookie传递信息的时候，大量cookie占用带宽；

2 Session在多台服务器之间的数据共享，广播数据容易形成网络风暴 ；

### Session 如何基于 Cookie 来工作

第一种情况下，当浏览器不支持 Cookie 功能时，浏览器会将用户的 SessionCookieName 重写到用户请求的 URL 参数中，它的传递格式如 /path/Servlet;name=value;name2=value2? Name3=value3，其中“Servlet；”后面的 K-V 对就是要传递的 Path Parameters，服务器会从这个Path Parameters 中拿到用户配置的SessionCookieName。关于这个 SessionCookieName，如果你在 web.xml 中配置 session-config 配置项的话，其 cookie-config 下的 name 属性就是这个 SessionCookieName 值，如果你没有配置 session-config 配置项，默认的 SessionCookieName 就是大家熟悉的“JSESSIONID”。接着 Request 根据这个 SessionCookieName 到Parameters 拿到Session ID并设置到 request.setRequestedSessionId 中。

请注意如果客户端也支持 Cookie 的话，Tomcat 仍然会解析 Cookie 中的 Session ID，并会覆盖 URL 中的 Session ID。

有了 Session ID 服务器端就可以创建 HttpSession 对象了，第一次触发是通过 request. getSession() 方法，如果当前的 Session ID 还没有对应的 HttpSession 对象那么就创建一个新的，并将这个对象加到 org.apache.catalina. Manager 的 sessions 容器中保存，Manager 类将管理所有 Session 的生命周期，Session 过期将被回收，服务器关闭，Session 将被序列化到磁盘等。只要这个 HttpSession 对象存在，用户就可以根据 Session ID 来获取到这个对象，也就达到了状态的保持。

### Cookie

#### Cookie的基概念

|  |
| --- |
| Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server. A cookie's value can uniquely identify a client, so cookies are commonly used for session management.  在servlet中创建一个Cookie，并且发送少量的信息到web浏览器，由浏览器保存这些信息，稍后再返回给服务器，cookie的值可以唯一的标识一个客户机  A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number. Some Web browsers have bugs in how they handle the optional attributes, so use them sparingly to improve the interoperability of your servlets.  cookie有一个名称、一个值和可选属性，如注释、路径和域限定符、最大年龄和版本号。一些浏览器有错误如何处理的可选属性，所以少量地使用它们来提高您的servlet的互操作性。 |

#### Cookie的创建

**[Cookie](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "Cookie(java.lang.String, java.lang.String))**(java.lang.String name, java.lang.String value) 构造参数  
          Constructs a cookie with a specified name and value.

#### Cookie的常用方法

|  |  |
| --- | --- |
| int | **[getMaxAge](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "getMaxAge())**()            Returns the maximum age of the cookie, specified in seconds, By default, -1 indicating the cookie will persist until browser shutdown. |
| java.lang.String | **[getName](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "getName())**()            Returns the name of the cookie. |
| java.lang.String | **[getPath](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "getPath())**()            Returns the path on the server to which the browser returns this cookie. |
| java.lang.String | **[getValue](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "getValue())**()            Returns the value of the cookie. |
| int | **[getVersion](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "getVersion())**()            Returns the version of the protocol this cookie complies with. |
| void | **[setMaxAge](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "setMaxAge(int))**(int expiry)            Sets the maximum age of the cookie in seconds. |
| void | **[setPath](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "setPath(java.lang.String))**(java.lang.String uri)            Specifies a path for the cookie to which the client should return the cookie. |
| void | **[setValue](mk:@MSITStore:D:/jsdk23/javax/servlet/http/Cookie.html" \l "setValue(java.lang.String))**(java.lang.String newValue)            Assigns a new value to a cookie after the cookie is created. |

#### Cookie中的不支持中文

 中文和英文字符不同，中文属于Unicod字符，在内存中站4个字符，而英文属于ASCII字符，内存中只占2个字符。Cookie中使用Unicode字符时需要对Unicode字符进行编码，否则会乱码。编码使用java.net.URLEncoder类的encode（String str,String encoding）方法，解码使用java.net.URLDecoder类的decode(String str,String encoding)方法。

当存储中文的时候要将cookie发送到浏览器时会报数据非法异常，所以要先编码，再解码

### Session

识别唯一的一个用户，并且为这个用户存储一些会话时产生的信息，保存在服务器端 。

作为一次会话范围的作用域 ，存储数据 。

#### Session的基本概念

servlet容器使用此接口在HTTP客户机和HTTP服务器之间创建会话。会话在指定的时间段内持续，跨越用户的多个连接或页面请求。会话通常对应于一个可以多次访问站点的用户。服务器可以通过多种方式维护会话，比如使用cookie或重写URL

#### 怎么获取session对象

|  |  |
| --- | --- |
| [HttpSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html) | **[getSession](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession())**() HttpServletRequest对象的方法           Returns the current session associated with this request, or if the request does not have a session, creates one. |

#### 常用方法

|  |  |
| --- | --- |
| java.lang.Object | **[getAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getAttribute(java.lang.String))**(java.lang.String name)            Returns the object bound with the specified name in this session, or null if no object is bound under the name. |
| java.util.Enumeration | **[getAttributeNames](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getAttributeNames())**()            Returns an Enumeration of String objects containing the names of all the objects bound to this session. |
| java.lang.String | **[getId](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getId())**()            Returns a string containing the unique identifier assigned to this session. |
| int | **[getMaxInactiveInterval](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getMaxInactiveInterval())**()            Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses. |
| java.lang.Object | **[getValue](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getValue(java.lang.String))**(java.lang.String name)            **Deprecated.** *As of Version 2.2, this method is replaced by [getAttribute(java.lang.String)](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getAttribute(java.lang.String)).* |
| java.lang.String[] | **[getValueNames](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getValueNames())**()            **Deprecated.** *As of Version 2.2, this method is replaced by [getAttributeNames()](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "getAttributeNames())* |
| void | **[putValue](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "putValue(java.lang.String, java.lang.Object))**(java.lang.String name, java.lang.Object value)            **Deprecated.** *As of Version 2.2, this method is replaced by [setAttribute(java.lang.String, java.lang.Object)](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "setAttribute(java.lang.String, java.lang.Object))* |
| void | **[removeAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "removeAttribute(java.lang.String))**(java.lang.String name)            Removes the object bound with the specified name from this session. |
| void | **[removeValue](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "removeValue(java.lang.String))**(java.lang.String name)            **Deprecated.** *As of Version 2.2, this method is replaced by [removeAttribute(java.lang.String)](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "removeAttribute(java.lang.String))* |
| void | **[setAttribute](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "setAttribute(java.lang.String, java.lang.Object))**(java.lang.String name, java.lang.Object value)            Binds an object to this session, using the name specified. |
| void | **[setMaxInactiveInterval](mk:@MSITStore:D:/jsdk23/javax/servlet/http/HttpSession.html" \l "setMaxInactiveInterval(int))**(int interval)            Specifies the time, in seconds, between client requests before the servlet container will invalidate this session. |