Servlet 处理日期

使用 Servlet 的最重要的优势之一是，可以使用 Core Java 中的大多数可用的方法。本章将讲解 Java 提供的 **java.util** 包中的**日期**类，这个类封装了当前的日期和时间。

Date 类支持两个构造函数。第一个构造函数初始化当前日期和时间的对象。

```
日期（）
```

下面的构造函数接受一个参数，该参数等于 1970 年 1 月 1 日午夜以来经过的较长数。

```
日期（长毫秒）
```

一旦您有一个可用的 Date 对象，您可以调用以下任意支持的方法来使用日期：

序号	方法 & 描述
1	**boolean after(Date date)** 如果调用的 Date 对象中包含的日期在 date 指定的日期之后，则返回 true，否则返回 false。
2	**boolean before(Date date)** 如果调用的 Date 对象中包含的日期在 date 指定的日期之前，则返回 true，否则返回 false。
3	**Object clone( )** 重复调用 Date 对象。
4	**int compareTo(Date date)** 把调用对象的值与 date 的值进行比较。如果两个值是相等的，则返回 0。如果调用对象在 date 之前，则返回一个负值。如果调用对象在 date 之后，则返回一个正值。
5	**int compareTo(Object obj)** 如果 obj 是 Date 类，则操作等同于 compareTo(Date)。否则，它会抛出一个 ClassCastException。
6	**boolean equals(Object date)** 如果调用的 Date 对象中包含的时间和日期与 date 指定的相同，则返回 true，否则返回 false。

7	long getTime( ) 返回 1970 年 1 月 1 日以来经过的毫秒数。
8	int hashCode( ) 为调用对象返回哈希代码。
9	void setTime(long time) 设置 time 指定的时间和日期，这表示从 1970 年 1 月 1 日午夜以来经过的时间（以毫秒为单位）。
10	String toString( ) 转换调用的 Date 对象为一个字符串，并返回结果。

获取当前的日期和时间

在 Java Servlet 中获取当前的日期和时间是非常容易的。您可以使用一个简单的 Date 对象的 *toString（）*方法来输出当前的日期和时间，如下所示：

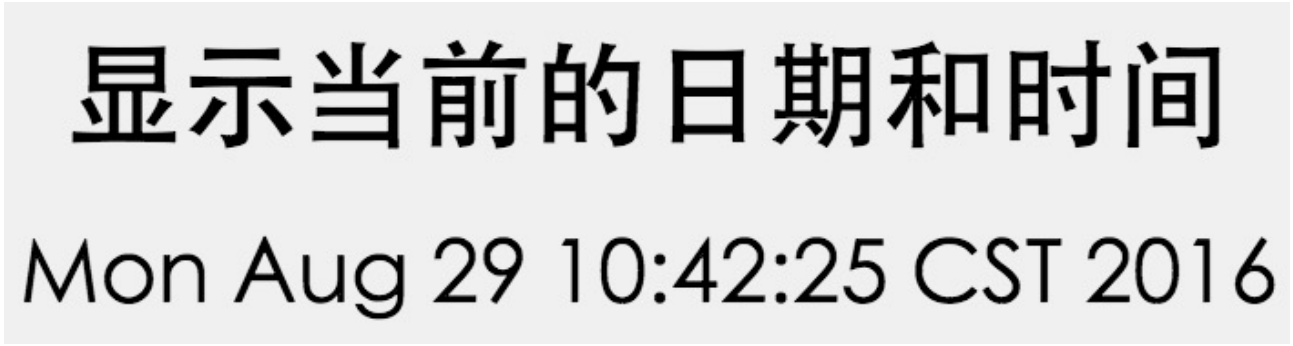
```
包 com.runoob.test;

导入 java.io.IOException;
导入 java.io.PrintWriter;
导入 java.util.Date;

导入 javax.servlet.ServletException;
导入 javax.servlet.annotation.WebServlet;
导入 javax.servlet.http.HttpServlet;
导入 javax.servlet.http.HttpServletRequest;
导入 javax.servlet.http.HttpServletResponse;

/ **
 * Servlet 实现类 CurrentDate
 * /
@WebServlet（“ / CurrentDate”）
公共类 CurrentDate 扩展 HttpServlet {
    私有静态最终长 serialVersionUID = 1L;

    公共 CurrentDate（）{
        超（）;
```

```
    }

    受保护的 void doGet（HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletException，IOE
xception {受保护的 void doGet （HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletExcept
ion ， IOException {

        response.setContentType（“ text / html; charset = UTF-8”）;。setContentType （“ text / html
; charset = UTF-8” ）;
        PrintWriter out = response.getWriter（）;PrintWriter out = 响应。getWriter （）;
          字符串标题=“显示当前的日期和时间”;字符串标题= “显示当前的日期和时间” ;
        日期日期=新的 Date（）;日期日期= 新日期（）;
        字符串 docType =“ <! DOCTYPE html> \ n”;字符串 docType = “ <! DOCTYPE html> \ n” ;
          out.println（docType +出来。println （docType +
            “ <html> \ n” +“ <html> \ n” +
            “ <head> <title>” +标题+“ </ title> </ head> \ n” +“ <head> <title>” + 标题+ “ </ title
> </ head> \ n” +
            “ <body bgcolor = \”＃f0f0f0 \“> \ n” +“ <body bgcolor = \”＃f0f0f0 \“> \ n” +
            “ <h1 align = \” center \“>” +标题+“ </ h1> \ n” +“ <h1 align = \” center \“>” + 标题+ “
</ h1> \ n” +
            “ <h2 align = \” center \“>” + date.toString（）+“ </ h2> \ n” +“ <h2 align = \” center
\“>” + 日期。toString （）+ “ </ h2> \ n” +
            “ </ body> </ html>”）; “ </ body> </ html>” ）;
    }}

}}
```

现在，让我们来编译上面的 Servlet，并在 web.xml 文件中创建适当的压缩：

```
<？ xml 版本=“ 1.0”编码=“ UTF-8”？ >xml 版本= “ 1.0” 编码= “ UTF-8” ？ >
<网络应用程序><网络应用程序>
  <servlet><servlet>
    <servlet-name> CurrentDate </ servlet-name><servlet-name> CurrentDate </ servlet-name>
    <servlet-class> com.runoob.test.CurrentDate </ servlet-class><servlet-class> com.runoob.test.C
urrentDate </ servlet-class>
  </ servlet></ servlet>
  <servlet 映射><servlet 映射>
    <servlet-name> CurrentDate </ servlet-name><servlet-name> CurrentDate </ servlet-name>
    <url-pattern> / TomcatTest / CurrentDate </ url-pattern><url-pattern> / TomcatTest / CurrentDa
te </ url-pattern>
  </ servlet-mapping></ servlet-mapping>
</ web-app></ web-app>
```

然后通过访问 http：// localhost：8080 / TomcatTest / CurrentDate 来调用该 Servlet。这将会产生如下的结果：



尝试刷新 URL http：// localhost：8080 / TomcatTest / CurrentDate，每隔几秒刷新一次您都会发现显示时间的差异。

日期比较

正如上面所提到的，您可以在 Servlet 中使用所有可用的 Java 方法。如果您需要比较两个日期，以下是方法：

- 您可以使用 getTime（）来获取两个对象自 1970 年 1 月 1 日午夜以来经过的时间（以较长为单位），然后对这两个值进行比较。
- 通过一个月里 12 号在 18 号之前，例如，new Date（99，2，12）.before（new Date（99，2，18）。您可以使用方法 before（），after（）和 equals（）。 ）返回 true。
- 您可以使用 compareTo（）方法，该方法由 Comparable 接口定义，由 Date 实现。

使用 SimpleDateFormat 格式化日期

SimpleDateFormat 是一个以语言环境敏感的方式来格式化和解析日期的特定类。SimpleDateFormat 允许您选择任何用户定义的日期时间格式化的模式。

让我们修改上面的实例，如下所示：

```
包 com.runoob.test;com 。runoob 。测试;

导入 java.io.IOException;导入 java 。io 。IOException ;
导入 java.io.PrintWriter;导入 java 。io 。PrintWriter ;
导入 java.text.SimpleDateFormat;导入 java 。文字。SimpleDateFormat ;
导入 java.util.Date;导入 java 。实用程序。日期;

导入 javax.servlet.ServletException;导入 javax 。servlet 。ServletException ;
导入 javax.servlet.annotation.WebServlet;导入 javax 。servlet 。注解。WebServlet ;
导入 javax.servlet.http.HttpServlet;导入 javax 。servlet 。http 。HttpServlet ;
导入 javax.servlet.http.HttpServletRequest;导入 javax 。servlet 。http 。HttpServletRequest ;
```

```
导入 javax.servlet.http.HttpServletResponse;导入 javax 。servlet 。http 。HttpServletResponse ;

/ **/ **
 * Servlet 实现类 CurrentDate
 * /
@WebServlet（“ / CurrentDate”）@WebServlet （“ / CurrentDate” ）
公共类 CurrentDate 扩展 HttpServlet {公共类 CurrentDate 扩展 HttpServlet {
    私有静态最终长 serialVersionUID = 1L;私有静态最终长 serialVersionUID = 1L ;

    公共 CurrentDate（）{公共 CurrentDate （）{
        超（）;超级（）;
    }}

    受保护的 void doGet（HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletException，IOE
xception {受保护的 void doGet （HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletExcept
ion ， IOException {

        response.setContentType（“ text / html; charset = UTF-8”）;。setContentType （“ text / html
; charset = UTF-8” ）;

        PrintWriter out = response.getWriter（）;PrintWriter out = 响应。getWriter （）;
        字符串标题=“显示当前的日期和时间”;字符串标题= “显示当前的日期和时间” ;
        日期 dNow =新的 Date（）;日期 dNow = 新日期（）;
        SimpleDateFormat ft = SimpleDateFormat ft =
                新的 SimpleDateFormat（“ yyyy.MM.dd hh：mm：ss E a”）; 新的 SimpleDateFormat （“ yyy
y.MM.dd hh：mm：ss E a” ）;
        字符串 docType =“ <! DOCTYPE html> \ n”;字符串 docType = “ <! DOCTYPE html> \ n” ;
        out.println（docType +出来。println （docType +
            “ <html> \ n” +“ <html> \ n” +
            “ <head> <title>” +标题+“ </ title> </ head> \ n” +“ <head> <title>” + 标题+ “ </ title>
</ head> \ n” +
            “ <body bgcolor = \”＃f0f0f0 \“> \ n” +“ <body bgcolor = \”＃f0f0f0 \“> \ n” +
            “ <h1 align = \” center \“>” +标题+“ </ h1> \ n” +
            “ <h2 align = \” center \“>” + ft.format（dNow）+“ </ h2> \ n” +
            “ </ body> </ html>”）;
    }

}
```

再次编译上面的 Servlet，然后通过访问 http：// localhost：8080 / TomcatTest / CurrentDate 来调用该 Servlet。这将会产生如下的结果：

显示当前的日期和时间

2016.08.29 10:48:52 星期一 上午

简单的日期格式的格式代码

使用事件模式字符串来指定时间格式。在这种模式下，所有的 ASCII 字母被保留为模式字母，这些字母定义如下：

字符	描述	实例
G	时代指标	广告
ÿ	四位数表示的年	2001
中号	一年中的月	七月或 07
d	一月中的第几天	10
H	带有 AM / PM 的小时（1∽12）	12
H	一天中的第几小时（0∽23）	22
米	一小时中的第几分	30
s	一分中的第几秒	55
小号	几个月	234
Ë	一周中的星期几	星期二
d	一年中的第几天	360
F	所在的周是这个月的第几周	2（七月的第二个星期三）

w	一年中的第几周	40
w ^	一月中的第几周	1 个
一个	AM / PM 标记	下午
ķ	一天中的第几小时（1∽24）	24
ķ	带有 AM / PM 的小时（0∽11）	10
ž	时区	东部标准时间
'	转义文字	定界符
”	单引号	、

如果要查看可用的处理日期方法的完整列表，则可以参考标准的 Java 文档。