Date & Time Handling In JAVA

Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email:info@sisoft.in
Phone: +91-9999-283-283

Date and Time API

System.currentTimeMillis()	A static method that returns the current date and time as milliseconds since January 1st 1970
java.util.Date	A class that represents a date and time
java.util.Calendar	A base class for calendar classes. Has methods to do date and time arithmethics like adding a day or month to another date
java.text.SimpleDateFormat	A class that can help you parse String's into Date's, and format Date's as String's

java.util.Date

Java provides the **Date** class available in **java.util** package, this class encapsulates the current date and time.

The Date class supports two constructors:

1) **Date( )**

   This constructor initializes the object with the current date and time.

2) **Date(long millisec)**

   This constructor accepts an argument that equals the number of milliseconds that have elapsed since midnight, January 1, 1970

Getting Current Date and Time

This is a very easy method to get current date and time in Java. You can use a simple Date object with *toString()* method to print the current date and time as follows −

```
import java.util.Date;

 public class DateDemo
 {
public static void main(String args[])
{
// Instantiate a Date object
Date date = new Date(); // display time and date using toString()
System.out.println(date.toString());
}
}
```

Output :
on May 04 09:51:52 CDT 2009

Date Formatting Using SimpleDateFormat

SimpleDateFormat is a concrete class for formatting and parsing dates in a locale-sensitive manner. SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting.

```
import java.util.*;
import java.text.*;

public class DateDemo
{
 public static void main(String args[])
{
 Date dNow = new Date( );
 SimpleDateFormat ft = new SimpleDateFormat ("E yyyy.MM.dd 'at' hh:mm:ss a zzz");
 System.out.println("Current Date: " + ft.format(dNow));
 }
 }
```

Output :
Current Date: Sun 2004.07.18 at 04:14:09 PM PDT

Simple DateFormat Format Codes

To specify the time format, use a time pattern string. In this pattern, all ASCII letters are reserved as pattern letters, which are defined as the following –

Character	Description	Example
M	Month in year	July or 07
d	Day in month	10
h	Hour in A.M./P.M. (1~12)	12
H	Hour in day (0~23)	22
m	Minute in hour	30
s	Second in minute	55
S	Millisecond	234
E	Day in week	Tuesday
D	Day in year	360

Measuring Elapsed Time

Sometimes, we may need to measure point in time in milliseconds. See below Example:

```
import java.util.*;
public class DiffDemo
{
 public static void main(String args[])
{
 try
{
 long start = System.currentTimeMillis( );
 System.out.println(new Date( ) + "\n");

Thread.sleep(5*60*10);
System.out.println(new Date( ) + "\n");

long end = System.currentTimeMillis( );
long diff = end - start;
System.out.println("Difference is : " + diff);
}
```

```
catch (Exception e)
{
 System.out.println("Got an
exception!");
} } }
```

Output:

```
Sun May 03 18:16:51 GMT 2009
Sun May 03 18:16:57 GMT 2009
Difference is : 5993
```

Date & Time Another Examples

1. Example: Convert Date to String.

```
SimpleDateFormat sdf = new
SimpleDateFormat("dd/M/yyyy");
String date = sdf.format(new Date());
System.out.println(date); //15/10/2013
```

2. Example: Convert String to Date.

```
SimpleDateFormat sdf = new SimpleDateFormat("dd-M-yyyy hh:mm:ss");
String dateInString = "31-08-1982 10:20:56";
Date date = sdf.parse(dateInString);
System.out.println(date); //Tue Aug 31 10:20:56 SGT 1982
```

3. Example: Get current date time:

```
SimpleDateFormat d= new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
Date date = new Date();
System.out.println(d.format(date));          //2013/10/15 16:16:39
```

Calender Class

Introduction

The **java.util.calendar** class is an abstract class that provides methods for converting between a specific instant in time and a set of calendar fields such as YEAR, MONTH, DAY_OF_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.Following are the important points about Calendar:

This class also provides additional fields and methods for implementing a concrete calendar system outside the package.

Calendar defines the range of values returned by certain calendar fields.

Class constructors :

1) **protected Calendar()**

This constructor constructs a Calendar with the default time zone and locale.

2) **protected Calendar(TimeZone zone, Locale aLocale)**

This constructor constructs a calendar with the specified time zone and locale.

Calender Examples

Example: 1

Date to Calendar (setTime())

```
 DateFormat formatter = new SimpleDateFormat("yyyyMMdd");
date = (Date)formatter.parse(date.toString());
cal=Calendar.getInstance();
cal.setTime(date);
```

Example: 2

Calendar to Date (getTime())

```
Calendar cal = Calendar.getInstance();
cal.add(Calendar.DATE, 1);
Date date = cal.getTime();
SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd");
String date1 = format1.format(date);
```

Example: 2

Set a date manually

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy MMM dd HH:mm:ss");
Calendar calendar = new GregorianCalendar(2013,1,28,13,24,56);
System.out.println(sdf.format(calendar.getTime())); //update a date

calendar.set(Calendar.YEAR, 2014);
calendar.set(Calendar.MONTH, 11);
calendar.set(Calendar.MINUTE, 33);
System.out.println("#2. " + sdf.format(calendar.getTime()));
```

Output :

2013 Feb 28 13:24:56 #2. 2014 Dec 28 13:33:56