

How do you format date and time in Android?

[Ask Question](#)

281

How do you format correctly according to the device configuration a date and time when having year, month, day, hour and minute?



[android](#) [date](#) [time](#) [formatting](#) [format](#)



71

asked Jan 17 '09 at 23:41

[pupeno](#)

108k

100

277

452

In Android, DateTime and Date parsing with example
androidwave.com/format-datetime-in-android –
[Surya Prakash Kushawah](#) Dec 5 '18 at 14:25

22 Answers



272

Use the standard Java DateFormat class.

For example to display the current date and time do the following:



```
Date date = new Date(location.getTime());
DateFormat dateFormat =
    android.text.format.DateFormat.getDateFormat(getApplicatio
mTimeText.setText("Time: " + dateFormat.format(date));
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

been deprecated and you should really be using a Java Calendar object.

edited Oct 1 '10 at 9:12



pupeno

108k 100 277 452

answered Jan 19 '09 at 8:57



JamieH

3,748 5 23 25

39 This is the `android.text.format.DateFormat` rather than `java.text.DateFormat`. – [jamesh](#) Sep 7 '09 at 23:31

4 It's pretty typical of Android IME to have two classes that both claim to give you a result that is set to the default Locale but one doesn't. So yes, don't forget to use the `android.text.format` version of `DateFormat` (that doesn't even derive the `java.util` one LOL). – [mxcl](#) Jul 20 '10 at 14:14

22 Please note this line: `DateFormat dateFormat = android.text.format.DateFormat.getDateFormat(getApplicationContext());` The returned `dateFormat` is of type `java.text.DateFormat` (and NOT `android.text.format.DateFormat`) – [Arye Rosenstein](#) Feb 13 '11 at 6:53

2 @Harsha - to get around that issue, I chain my use of `DateFormat` so I only have to reference the Android class and therefore there aren't any ambiguous classes. `final String dateStr = DateFormat.getDateFormat(this).format(d);` You can use Android's `format()` method and have (IMHO) cleaner code and one less Object to instantiate. – [Jerry Brady](#) Aug 22 '11 at 19:06

19 This formatter only includes the date, not the time as the original question stated. Use `DateUtils` from the same package instead, see stackoverflow.com/questions/2983920/ – [Asmo Soinio](#) Dec 2 '11 at 13:43

Home

PUBLIC

Stack Overflow

Tags

Users

...

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Teams

Q&A for work


[Learn More](#)

186

In my opinion,

```
android.text.format.DateFormat.getDateFormat(context)
makes me confused because this method returns
java.text.DateFormat rather than
android.text.format.DateFormat - "-".
```

So, I use the fragment code as below to get the current date/time in my format.

```
android.text.format.DateFormat df = new android.text.forma
df.format("yyyy-MM-dd hh:mm:ss a", new java.util.Date());
```

or

```
android.text.format.DateFormat.format("yyyy-MM-dd hh:mm:ss
```

In addition, you can use others formats. Follow [DateFormat](#).

edited Dec 9 '16 at 11:42



[ABM Ruman](#)

69 1 8

answered Oct 14 '10 at 4:34



[Fuangwith S.](#)

3,509 7 31 39

20 Useful, but the question said "according to the device configuration". To me that implies using a format chosen based on the user's language/country, or chosen directly by the user, rather than hardcoding the choice of format. – [Chris Boyle](#) Oct 14 '10 at 16:26

14 also, don't forget that hh:mm:ss will give you 01:00:00 for 1 PM, you'll need to use kk:mm:ss to get 13:00:00 – [dnet](#) Apr 17 '12 at 11:51

2 @dnet k is hour in day (1-24), do you not mean H , which is hour in dav (0-23). eg. HH:mm:ss? See:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

1 @Joony no, there's difference between
`java.text.SimpleDateFormat` (what you linked and uses `H` for hours in the 0-23 range) and
`android.text.format.DateFormat` (what the answer is about and uses `k` for hours in the 0-23 range) – [dnet](#) Dec 7 '12 at 22:51

@dnet After testing, you are correct about `k`, however, the documentation for `DateFormat` clearly states *For the canonical documentation of format strings, see `SimpleDateFormat`*. Very confusing. Or am I missing something? – [Joony](#) Dec 10 '12 at 11:24



You can use `DateFormat`. Result depends on default Locale of the phone, but you can specify Locale too :

93



<https://developer.android.com/reference/java/text/DateFormat.html>

This is results on a

```
DateFormat.getInstance().format(date)
```

FR Locale : 3 nov. 2017

US/En Locale : Jan 12, 1952

```
DateFormat.getInstance(DateFormat.SHORT).format(date)
```

FR Locale : 03/11/2017

US/En Locale : 12.13.52

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

FR Locale : 3 nov. 2017

US/En Locale : Jan 12, 1952

```
DateFormat.getDateInstance(DateFormat.LONG).format(date)
```

FR Locale : 3 novembre 2017

US/En Locale : January 12, 1952

```
DateFormat.getDateInstance(DateFormat.FULL).format(date)
```

FR Locale : vendredi 3 novembre 2017

US/En Locale : Tuesday, April 12, 1952

```
DateFormat.getDateTimeInstance().format(date)
```

FR Locale : 3 nov. 2017 16:04:58

```
DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.SHORT).format(date)
```

FR Locale : 03/11/2017 16:04

```
DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.SHORT).format(date)
```

FR Locale : 03/11/2017 16:04:58

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

FR Locale : 03/11/2017 16:04:58 GMT+01:00

```
DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat
```

FR Locale : 03/11/2017 16:04:58 heure normale d'Europe centrale

```
DateFormat.getTimeInstance().format(date)
```

FR Locale : 16:04:58

```
DateFormat.getTimeInstance(DateFormat.SHORT).format(date)
```

FR Locale : 16:04

```
DateFormat.getTimeInstance(DateFormat.MEDIUM).format(date)
```

FR Locale : 16:04:58

```
DateFormat.getTimeInstance(DateFormat.LONG).format(date)
```

FR Locale : 16:04:58 GMT+01:00

```
DateFormat.getTimeInstance(DateFormat.FULL).format(date)
```

FR Locale : 16:04:58 heure normale d'Europe centrale

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

**Dario Bruzzese****1,159** 12 13

answered Nov 3 '17 at 15:09

**Dany Poplawec****2,165** 1 16 24

-
- 2 Thank you for putting all these cases in one place. If you could add cases for time only as well, this will make it a complete reference. – [zeeshan](#) Nov 21 '17 at 19:05
-

Simple Date format with example in Android
androidwave.com/format-datetime-in-android –
[Surya Prakash Kushawah](#) Dec 5 '18 at 14:28



Date to Locale date string:

36

```
Date date = new Date();
String stringDate = DateFormat.getDateInstance().format(date);
```



Options:

```
DateFormat.getDateInstance()
```

- > Dec 31, 1969

```
DateFormat.getDateTimeInstance()
```

-> Dec 31, 1969 4:00:00 PM

```
DateFormat.getTimeInstance()
```



Ivo Stoyanov

10.8k 5 39 43

- 3 How to remove year from DateFormat.getDateInstance() ? –
[Hardik Joshi](#) Nov 8 '15 at 17:36

Here is most of example androidwave.com/format-datetime-in-android – [Surya Prakash Kushawah](#) Dec 5 '18 at 14:28



This will do it:

19



```
Date date = new Date();
java.text.DateFormat dateFormat =
android.text.format.DateFormat.getDateFormat(getApplication()
mTimeText.setText("Time: " + dateFormat.format(date));
```

edited Nov 25 '16 at 10:22



Maveñ

7,284 12 43 78

answered Aug 3 '10 at 21:48



tronman

6,426 7 34 41

Will this get localized when user changes the language in an
android device? – [Kannan_SJD](#) Jul 1 '16 at 7:16




Following this:

<http://developer.android.com/reference/android/text/format/Time.html>

1 1

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



```
Time now = new Time();
now.setToNow();
```

Then format:

```
Log.d("DEBUG", "Time "+now.format("%d.%m.%Y %H.%M.%S"));
```

answered Jan 5 '14 at 16:50



[FireZenk](#)

664 1 14 25

-
- 1 @FireZenk: According to the [link](developer.android.com/reference/android/text/format/Time.html) you provided: *This class has a number of issues and it is recommended that GregorianCalendar is used instead.* – [ccpizza](#) Jan 1 '15 at 14:30
-


Oh... this issue-info is newer than my comment, so that's a deprecated answer – [FireZenk](#) Jan 2 '15 at 14:59

Use SimpleDateFormat

Like this:

13

```
event.putExtra("starttime", "12/18/2012");
```



```
SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy");
Date date = format.parse(bundle.getString("starttime"));
```

answered Dec 18 '12 at 7:55



[neknek mouh](#)

1,007 5 22 51

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

iutinv Apr 10 '13 at 5:33 

Use these two as a class variables:

8

```
public java.text.DateFormat dateFormat = new SimpleDateFormat();
private Calendar mDate = null;
```



And use it like this:

```
mDate = Calendar.getInstance();
mDate.set(year, months, day);
dateFormat.format(mDate.getTime());
```

edited Jul 3 '12 at 17:23



Peter Mortensen

14.1k 19 88 114

answered Jun 23 '12 at 5:44

user1367623



This is my method, you can define and input and output format.

8



```
public static String formattedDateFromString(String inputFormat,
String inputDate){
    if(inputFormat.equals("")){ // if inputFormat = "", set default
        inputFormat = "yyyy-MM-dd hh:mm:ss";
    }
    if(outputFormat.equals("")){
        outputFormat = "EEEE d 'de' MMMM 'del' yyyy" // if i
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
String outputDate = "";

SimpleDateFormat df_input = new SimpleDateFormat(inputFormat,
    java.util.Locale.getDefault());
SimpleDateFormat df_output = new SimpleDateFormat(outputFormat,
    java.util.Locale.getDefault());

// You can set a different Locale, This example set a
//SimpleDateFormat df_input = new SimpleDateFormat(inputFormat,
//    java.util.Locale.getDefault());
//SimpleDateFormat df_output = new SimpleDateFormat(outputFormat,
//    java.util.Locale.getDefault());

try {
    parsed = df_input.parse(inputDate);
    outputDate = df_output.format(parsed);
} catch (Exception e) {
    Log.e("formattedDateFromString", "Exception in formatting date from string", e);
}
return outputDate;
}
```

answered Apr 11 '14 at 17:54



Elenasys

96.8k 16 253 217



SimpleDateFormat

8



I use SimpleDateFormat **without custom pattern** to get actual date and time from the system in **the device's preselected** format:

```
public static String getFormattedDate() {
    //SimpleDateFormat called without pattern
    return new SimpleDateFormat().format(Calendar.getInstance().getTime());
}
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

returns:

- 13.01.15 11:45
- 1/13/15 10:45 AM
- ...

edited Aug 16 '17 at 9:25

answered Jan 13 '15 at 10:55



Tomas

2,138 4 22 35



Use build in Time class!

7

```
Time time = new Time();
time.set(0, 0, 17, 4, 5, 1999);
Log.i("DateTime", time.format("%d.%m.%Y %H:%M:%S"));
```



answered Jan 16 '13 at 13:10



Igor Krumpak

165 2 1

This is the optimal solution because it uses the lightweight Time object from the Android Framework:
developer.android.com/reference/android/text/format/Time.html
! – Dean Wild Jan 29 '13 at 12:17

- 13 This is **not** the optimal solution: it doesn't respect the date format from the user's locale. – Dan Hulme Apr 25 '13 at 16:47

▲ This code would return the current date and time:

4

```
public String getCurrDate()
{
    String dt;
    Date cal = Calendar.getInstance().getTime();
    dt = cal.toLocaleString();
    return dt;
}
```

edited Jul 3 '12 at 17:21



Peter Mortensen

14.1k 19 88 114

answered Jul 27 '11 at 10:39



chetan

83 1 2

toLocaleString() is deprecated – [Kopfgeldjaeger](#) Mar 28 '13 at 17:13

▲ The other answers are generally correct. I should like to contribute the modern answer. The classes `Date`, `DateFormat` and `SimpleDateFormat` used in most of the other answers, are long outdated and have caused trouble for many programmers over many years. Today we have so much better in `java.time`, AKA JSR-310, the modern Java date & time API. Can you use this on Android yet? Most certainly! The modern classes have been backported to Android in the ThreeTenABP project. See [this question: How to use ThreeTenABP in Android Project](#) for all the details.

4

▲ This snippet should get you started:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
int year = 2017, month = 9, day = 28, hour = 22, minute = 45;
LocalDateTime dateTime = LocalDateTime.of(year, month, day, hour, minute);
DateTimeFormatter formatter =
    DateTimeFormatter.ofLocalizedDateTime(FormatStyle.MEDIUM);
System.out.println(dateTime.format(formatter));
```

When I set my computer's preferred language to US English or UK English, this prints:

```
Sep 28, 2017 10:45:00 PM
```

When instead I set it to Danish, I get:

```
28-09-2017 22:45:00
```

So it does follow the configuration. I am unsure exactly to what detail it follows your device's date and time settings, though, and this may vary from phone to phone.

answered Sep 8 '17 at 9:03



Ole V.V.

33.4k 7 43 59



This code work for me!

4

```
Date d = new Date();
CharSequence s = android.text.format.DateFormat.format(
    "ss", d.getTime());
Toast.makeText(this, s.toString(), Toast.LENGTH_SHORT).show();
```



answered Oct 8 '17 at 18:04



Opriday

33.4k 7 43 59

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



Try:

2

```
event.putExtra("startTime", "10/05/2012");
```



And when you are accessing passed variables:

```
SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/y");
Date date = formatter.parse(bundle.getString("startTime"))
```

edited Jul 3 '12 at 17:22



Peter Mortensen

14.1k 19 88 114

answered Mar 26 '12 at 6:46



Abhishek Singh Rathaur

265 3 3



I use it like this:

2

```
public class DateUtils {
    static DateUtils instance;
    private final DateFormat dateFormat;
    private final DateFormat timeFormat;

    private DateUtils() {
        dateFormat =
        android.text.format.DateFormat.getDateFormat(MainApplicati
        timeFormat =
        android.text.format.DateFormat.getTimeFormat(MainApplicati
    }

    public static DateUtils getInstance() {
        if (instance == null) {
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
}  
  
public synchronized static String formatDateTime(long  
    long milliseconds = timestamp * 1000;  
    Date dateTime = new Date(milliseconds);  
    String date = getInstance().dateFormat.format(date);  
    String time = getInstance().timeFormat.format(date);  
    return date + " " + time;  
}  
}
```

answered Nov 6 '13 at 22:15



ViliusK

8,189 2 58 66



Avoid j.u.Date

1



The `Java.util.Date` and `.Calendar` and `SimpleDateFormat` in Java (and Android) are notoriously troublesome. Avoid them. They are so bad that Sun/Oracle gave up on them, supplanting them with the new `java.time` package in Java 8 (not in Android as of 2014). The new [java.time](#) was inspired by the [Joda-Time](#) library.

Joda-Time

[Joda-Time](#) does work in Android.

Search StackOverflow for "Joda" to find many examples and much discussion.

A tidbit of source code using Joda-Time 2.4.

Standard format.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

to the ISO 8601 standard.

Localized format.

```
String output = DateTimeFormat.forStyle( "FF" ).print( Date  
// Full (long) format localized for this user's language at
```

edited Aug 8 '14 at 19:47

answered Aug 8 '14 at 19:41



Basil Bourque

121k 32 408 573



Back to 2016, When I want to customize the format (not according to the device configuration, as you ask...) I usually use the string resource file:



in strings.xml:

```
<string name="myDateFormat"><xliff:g id="myDateFormat">%1$  
</string>
```

In Activity:

```
Log.d(TAG, "my custom date format: "+getString(R.string.myI
```

This is also useful with the release of the new [Date Binding Library](#).

So I can have something like this in layout file:

```

android:layout_width="wrap_content"
android:layout_height="0dp"
android:layout_weight="1"
android:padding="2dp"
android:text="@{@string/myDateFormat(vm.releaseDate)}"
tools:text="0000"
/>

```

And in java class:

```

MovieDetailViewModel vm = new MovieDetailViewModel();
vm.setReleaseDate(new Date());

```

answered Nov 28 '16 at 17:51



alexpfx

1,946 4 22 45



Shortest way:

1

```

// 2019-03-29 16:11
String.format("%1$tY-%<tm-%<td %<tR", Calendar.getInstance

```



%tR is short for %tH:%tM , < means to reuse last parameter(1\$).

It is equivalent to String.format("%1\$tY-%1\$tm-%1\$td
%1\$tH:%1\$tM", Calendar.getInstance())

<https://developer.android.com/reference/java/util/Formatter.html>

answered Mar 29 at 9:11



Khoa Chu Anh

138 1 7

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



The android Time class provides 3 formatting methods
<http://developer.android.com/reference/android/text/format/Time.html>



This is how I did it:

```
/**
 * This method will format the data from the android Time c
 * into the format
 * Date: dd.mm.yy Time: hh.mm.ss
 */
private String formatTime(String time)
{
    String fullTime= "";
    String[] sa = new String[2];

    if(time.length()>1)
    {
        Time t = new Time(Time.getCurrentTimezone());
        t.parse(time);
        // or t.setToNow();
        String formattedTime = t.format("%d.%m.%Y %H.%M.%S");
        int x = 0;

        for(String s : formattedTime.split("\\s",2))
        {
            System.out.println("Value = " + s);
            sa[x] = s;
            x++;
        }
        fullTime = "Date: " + sa[0] + " Time: " + sa[1];
    }
    else{
        fullTime = "No time data";
    }
    return fullTime;
}
```

I hope thats helpful :-)

answered Aug 30 '13 at 9:22

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



It's too late but it may help to someone

0

```
DateFormat.format(format, timeInMillis);
```



here `format` is what format you need

ex: "HH:mm" returns 15:30

answered Oct 12 '16 at 10:38



[subrahmanyam boyapati](#)

1,951 10 25



Locale

0

To get date or time in locale format from milliseconds I used this:



Date and time

```
Date date = new Date(milliseconds);
DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.SHORT, Locale.getDefault());
dateFormat.format(date);
```

Date

```
Date date = new Date(milliseconds);
DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.SHORT, Locale.getDefault());
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Time

```
Date date = new Date(milliseconds);
DateFormat dateFormat = DateFormat.getInstance(DateFormat
Locale.getDefault());
dateFormat.format(date);
```

You can use other date style and time style. More info about styles [here](#).

answered Sep 26 '17 at 19:05



Wojtek

582 2 7 17

protected by [Community](#) ♦ Jul 24 '17 at 7:15

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).