**Java Dates SFD**

[1. Fuseau horaire (Time zone) et Temps universel coordonné UTC 2](#_Toc107439136)

[1.1. Carte des fuseaux horaires 2](#_Toc107439137)

[1.2. Décalage du fuseau horaire par rapport à UTC 2](#_Toc107439138)

[1.2.1. Pour une liste des décalages UTC 3](#_Toc107439139)

[1.2.2. Pour une liste **exhaustive** des décalages UTC 3](#_Toc107439140)

[2. Package java.time 3](#_Toc107439141)

[2.1. Classes du Package java.time 3](#_Toc107439142)

[3. ZoneID 4](#_Toc107439143)

[3.1. Utilisation de ZoneId 5](#_Toc107439144)

[3.1.1. Type1: Region-based ZoneId 6](#_Toc107439145)

[3.1.1.1. Application 6](#_Toc107439146)

[4. Format des dates 7](#_Toc107439147)

[4.1. Class SimpleDateFormat 7](#_Toc107439148)

[4.1.1. Date and Time Patterns 7](#_Toc107439149)

[4.1.2. Exemples 8](#_Toc107439150)

[4.2. Class DateTimeFormatter 9](#_Toc107439151)

[4.2.1. Predefined Formatters 9](#_Toc107439152)

[4.2.2. Patterns for Formatting and Parsing 10](#_Toc107439153)

[5. Liste déroulante 11](#_Toc107439154)

[5.1. Format des Dates 11](#_Toc107439155)

[6. JTabbedPane 13](#_Toc107439156)

[7. Links 16](#_Toc107439157)

# Fuseau horaire (Time zone) et Temps universel coordonné UTC

## Carte des fuseaux horaires

Une image contenant carte

Description générée automatiquement

Carte Fuseaux horaires = L’heure dans le monde par rapport à UTC.

[[1]](#footnote-1)

## Décalage du fuseau horaire par rapport à UTC

De façon simple, un fuseau horaire peut être écrit sous la **forme UTC+X ou UTC-Y, où « X » et « Y »** **représentent le décalage du fuseau par rapport à UTC[[2]](#footnote-2).**

Les exemples suivants donnent des exemples de variation de l'heure locale suivant plusieurs fuseaux horaires lorsqu'il est 12:00 UTC et en hiver :

Los Angeles, États-Unis (UTC-8) : 04:00

Chicago, États-Unis (UTC-6) : 06:00

New York, États-Unis (UTC-5) : 07:00

Halifax, Canada (UTC-4) : 08:00

Londres, Royaume-Uni (UTC±0) : 12:00

Stockholm, Suède (UTC+1) : 13:00

Le Cap, Afrique du Sud (UTC+2) : 14:00

Mysore, Inde (UTC+5:30) : 17:30

Katmandou, Népal (UTC+5:45) : 17:45

Séoul, Corée du Sud (UTC+9) : 21:00

Melbourne, Australie (UTC+10) : 22:00

### Pour une liste des décalages UTC

<https://en.wikipedia.org/wiki/Time_zone#List_of_UTC_offsets>

### Pour une liste **exhaustive** des décalages UTC

<https://en.wikipedia.org/wiki/List_of_UTC_offsets>

# Package java.time

L'API principale pour les dates, les heures, les instants et les durées.

Les classes définies ici représentent les principaux concepts date-heure, y compris les instants, les durées, les dates, les heures, les **fuseaux horaires** et les périodes.

Ils sont basés sur le système de calendrier ISO, qui est le calendrier mondial de facto suivant les règles proleptiques grégoriennes. Toutes les classes sont immuables et thread-safe.

## Classes du Package java.time

|  |  |  |
| --- | --- | --- |
| **Id** | **Class** | **Description** |
| 1 | ***Clock*** | A clock providing access to the current instant, date and time using a time-zone. |
| 2 | ***Duration*** | A time-based amount of time, such as '34.5 seconds'. |
| 3 | ***Instant*** | An instantaneous point on the time-line. |
| 4 | ***LocalDate*** | A date without a time-zone in the ISO-8601 calendar system, such as 2007-12-03. |
| 5 | ***LocalDateTime*** | A date-time without a time-zone in the ISO-8601 calendar system, such as 2007-12-03T10:15:30. |
| 6 | ***LocalTime*** | A time without a time-zone in the ISO-8601 calendar system, such as 10:15:30. |
| 7 | ***MonthDay*** | A month-day in the ISO-8601 calendar system, such as --12-03. |
| 8 | ***OffsetDateTime*** | A date-time with an offset from UTC/Greenwich in the ISO-8601 calendar system, such as 2007-12-03T10:15:30+01:00. |
| 9 | ***OffsetTime*** | A time with an offset from UTC/Greenwich in the ISO-8601 calendar system, such as 10:15:30+01:00. |
| 10 | ***Period*** | A date-based amount of time in the ISO-8601 calendar system, such as '2 years, 3 months and 4 days'. |
| 11 | ***Year*** | A year in the ISO-8601 calendar system, such as 2007. |
| 12 | ***YearMonth*** | A year-month in the ISO-8601 calendar system, such as 2007-12. |
| 13 | ***ZonedDateTime*** | A date-time with a time-zone in the ISO-8601 calendar system, such as 2007-12-03T10:15:30+01:00 Europe/Paris. |
| 14 | ***ZoneId*** | A time-zone ID, such as Europe/Paris. |
| 15 | ***ZoneOffset*** | A time-zone offset from Greenwich/UTC, such as +02:00. |

# ZoneID

**Class ZoneId**

public **abstract** class ZoneId

extends [Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)

implements [Serializable](https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html)

A time-zone ID, such as Europe/Paris[[3]](#footnote-3).

|  |  |  |
| --- | --- | --- |
| **Id** | **Modifier and Type** | **Method and Description** |
| 1 | boolean | ***equals***(Object obj)  Checks if this time-zone ID is equal to another time-zone ID. |
| 2 | static ZoneId | ***from***(TemporalAccessor temporal)  Obtains an instance of ZoneId from a temporal object. |
| 3 | static Set<String> | ***getAvailableZoneIds()***  Gets the set of available zone IDs. |
| 4 | String | ***getDisplayName(TextStyle style, Locale locale)***  Gets the textual representation of the zone, such as 'British Time' or '+02:00'. |
| 5 | abstract String | ***getId()***  Gets the unique time-zone ID. |
| 6 | abstract ZoneRules | ***getRules()***  Gets the time-zone rules for this ID allowing calculations to be performed. |
| 7 | int | ***hashCode()***  A hash code for this time-zone ID. |
| 8 | ZoneId | ***normalized()***  Normalizes the time-zone ID, returning a ZoneOffset where possible. |
| 9 | static ZoneId | ***of(String zoneId)***  Obtains an instance of ZoneId from an ID ensuring that the ID is valid and available for use. |
| 10 | static ZoneId | ***of(String zoneId, Map<String,String> aliasMap)***  Obtains an instance of ZoneId using its ID using a map of aliases to supplement the standard zone IDs. |
| 11 | static ZoneId | ***ofOffset(String prefix, ZoneOffset offset)***  Obtains an instance of ZoneId wrapping an offset. |
| 12 | static ZoneId | ***systemDefault()***  Gets the system default time-zone. |
| 13 | String | ***toString()***  Outputs this zone as a String, using the ID. |

Tableau Méthodes de Class ZoneId[[4]](#footnote-4)

## Utilisation de ZoneId

La classe ZoneId est utilisée pour identifier un fuseau horaire et fournir les règles de conversion entre ***LocalDateTime*** et ***Instant***. En termes de **règles de décalage (offset rules),** ZoneId est divisée en 2 types :

1. ZoneId avec un décalage de fuseau horaire fixe, tel que "UTC+07", "GMT-05:40", "UT-03", "+05:50".
2. ZoneId avec un décalage de fuseau horaire non fixe, tel que "Europe/Paris". Son décalage de fuseau horaire dépend de l'heure sur la chronologie ou dépend du jour de l'année[[5]](#footnote-5).

Bien que ***ZoneId*** soit une classe **abstraite** :

public **abstract** class ZoneId implements Serializable {

public abstract String getId();

public abstract ZoneRules getRules();

...

}

elle fournit quelques méthodes de fabrique **statiques** pour créer des objets ***ZoneId***.

Deux propriétés importantes de ***ZoneId*** sont ***id*** et ***rules***:

* ***String id***: l'ID est unique.
* ***ZoneRules rules***: Les règles permettant de déterminer le décalage de fuseau horaire à un moment précis de la chronologie.

La classe ZoneOffset est une sous-classe de ZoneId :

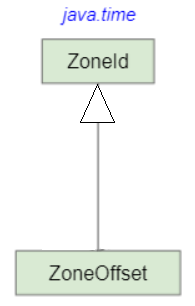


Figure ZoneOffset sous-classe de ZoneId

Basée sur la syntaxe de l'ID, ***ZoneId*** est divisée en 3 types :

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | | **Example** | **getId()** |
| **Type1** | **Region-based ZoneId** | **ZoneId**.of("Europe/Paris") | Europe/Paris |
| **Type2** | **Offset-based ZoneId** | **ZoneOffset**.of("-06") | -06 |
| **ZoneOffset**.of("+06:05:20") | +06:05:20 |
| **Type3** | **UTC/GMT/UT ZoneId** | **ZoneId**.ofOffset("UTC", **ZoneOffset**.of("+06")) | UTC+06 |
| **ZoneId**.of("GMT-06:05:20") | GMT-06:05:20 |

Tableau Les 3 typesde ZoneId

### Type1: Region-based ZoneId

**Region-based ZoneId** (**ZoneId** basée sur la région), la valeur du paramètre **zoneId** doit comporter 2 caractères ou plus et ne doit pas commencer par **"UTC", "GMT", "UT", "+", "- "**. De nombreuses valeurs sont fournies, telles que **"Europe/Paris", "Asia/Ho\_Chi\_Minh"**,...

La méthode statique **ZoneId.getAvailableZoneIds()** de la classe ZoneId renvoie un ensemble de ces **zoneId**(s). Cet ensemble comprend tous les ID disponibles basés sur la région.

* Cette méthode n’accepte aucun paramètre.
* Cette méthode renvoie des ***Set*** qui sont une copie modifiable de l’ensemble des ID de zone.

#### Application

public void allZonesId() {

Set<String> allZones = ZoneId.getAvailableZoneIds();

// create ArrayList from Set

List<String> zoneList = new ArrayList<String>(allZones);

Collections.sort(zoneList);

// printing first 10 zoneid with offset

//Obtains the current date-time from the system clock in the default time-zone.

LocalDateTime dtLoc = LocalDateTime.now();

for (int i = 0; i < 10; i++) {

// get zoneid then ZonedDateTime and offset from that ZoneId

String sItem = zoneList.get(i);

ZoneId meridianId = ZoneId.of(sItem);

ZonedDateTime zonedDate = dtLoc.atZone(meridianId);

ZoneOffset offset = zonedDate.getOffset();

System.out.println("ZoneId = " + meridianId + " offset = " + offset );

}

}

ZoneId = Africa/Abidjan offset = Z

ZoneId = Africa/Accra offset = Z

ZoneId = Africa/Addis\_Ababa offset = +03:00

ZoneId = Africa/Algiers offset = +01:00

ZoneId = Africa/Asmara offset = +03:00

ZoneId = Africa/Asmera offset = +03:00

ZoneId = Africa/Bamako offset = Z

ZoneId = Africa/Bangui offset = +01:00

ZoneId = Africa/Banjul offset = Z

ZoneId = Africa/Bissau offset = Z

# Format des dates

## Class SimpleDateFormat

### Date and Time Patterns

<https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Date or Time Component** | **Presentation** | **Examples** |
| G | Era designator | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | AD |
| y | Year | [Year](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#year) | 1996; 96 |
| Y | Week year | [Year](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#year) | 2009; 09 |
| M | Month in year | [Month](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#month) | July; Jul; 07 |
| w | Week in year | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 27 |
| W | Week in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 2 |
| D | Day in year | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 189 |
| d | Day in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 10 |
| F | Day of week in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 2 |
| E | Day name in week | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | Tuesday; Tue |
| u | Day number of week (1 = Monday, ..., 7 = Sunday) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 1 |
| a | Am/pm marker | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | PM |
| H | Hour in day (0-23) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 0 |
| k | Hour in day (1-24) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 24 |
| K | Hour in am/pm (0-11) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 0 |
| h | Hour in am/pm (1-12) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 12 |
| m | Minute in hour | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 30 |
| s | Second in minute | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 55 |
| S | Millisecond | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 978 |
| z | Time zone | [General time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#timezone) | Pacific Standard Time; PST; GMT-08:00 |
| Z | Time zone | [RFC 822 time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#rfc822timezone) | -0800 |
| X | Time zone | [ISO 8601 time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#iso8601timezone) | -08; -0800; -08:00 |

### Exemples

|  |  |
| --- | --- |
| **Date and Time Pattern** | **Result** |
| "yyyy.MM.dd G 'at' HH:mm:ss z" | 2001.07.04 AD at 12:08:56 PDT |
| "EEE, MMM d, ''yy" | Wed, Jul 4, '01 |
| "h:mm a" | 12:08 PM |
| "hh 'o''clock' a, zzzz" | 12 o'clock PM, Pacific Daylight Time |
| "K:mm a, z" | 0:08 PM, PDT |
| "yyyyy.MMMMM.dd GGG hh:mm aaa" | 02001.July.04 AD 12:08 PM |
| "EEE, d MMM yyyy HH:mm:ss Z" | Wed, 4 Jul 2001 12:08:56 -0700 |
| "yyMMddHHmmssZ" | 010704120856-0700 |
| "yyyy-MM-dd'T'HH:mm:ss.SSSZ" | 2001-07-04T12:08:56.235-0700 |
| "yyyy-MM-dd'T'HH:mm:ss.SSSXXX" | 2001-07-04T12:08:56.235-07:00 |
| "YYYY-'W'ww-u" | 2001-W27-3 |

## Class DateTimeFormatter

<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>

### Predefined Formatters

| **Formatter** | **Description** | **Example** |
| --- | --- | --- |
| [ofLocalizedDate(dateStyle)](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ofLocalizedDate-java.time.format.FormatStyle-) | Formatter with date style from the locale | '2011-12-03' |
| [ofLocalizedTime(timeStyle)](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ofLocalizedTime-java.time.format.FormatStyle-) | Formatter with time style from the locale | '10:15:30' |
| [ofLocalizedDateTime(dateTimeStyle)](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ofLocalizedDateTime-java.time.format.FormatStyle-) | Formatter with a style for date and time from the locale | '3 Jun 2008 11:05:30' |
| [ofLocalizedDateTime(dateStyle,timeStyle)](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ofLocalizedDateTime-java.time.format.FormatStyle-) | Formatter with date and time styles from the locale | '3 Jun 2008 11:05' |
| [BASIC\_ISO\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#BASIC_ISO_DATE) | Basic ISO date | '20111203' |
| [ISO\_LOCAL\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_LOCAL_DATE) | ISO Local Date | '2011-12-03' |
| [ISO\_OFFSET\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_OFFSET_DATE) | ISO Date with offset | '2011-12-03+01:00' |
| [ISO\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_DATE) | ISO Date with or without offset | '2011-12-03+01:00'; '2011-12-03' |
| [ISO\_LOCAL\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_LOCAL_TIME) | Time without offset | '10:15:30' |
| [ISO\_OFFSET\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_OFFSET_TIME) | Time with offset | '10:15:30+01:00' |
| [ISO\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_TIME) | Time with or without offset | '10:15:30+01:00'; '10:15:30' |
| [ISO\_LOCAL\_DATE\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_LOCAL_DATE_TIME) | ISO Local Date and Time | '2011-12-03T10:15:30' |
| [ISO\_OFFSET\_DATE\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_OFFSET_DATE_TIME) | Date Time with Offset | 2011-12-03T10:15:30+01:00' |
| [ISO\_ZONED\_DATE\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_ZONED_DATE_TIME) | Zoned Date Time | '2011-12-03T10:15:30+01:00[Europe/Paris]' |
| [ISO\_DATE\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_DATE_TIME) | Date and time with ZoneId | '2011-12-03T10:15:30+01:00[Europe/Paris]' |
| [ISO\_ORDINAL\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_ORDINAL_DATE) | Year and day of year | '2012-337' |
| [ISO\_WEEK\_DATE](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_WEEK_DATE) | Year and Week | 2012-W48-6' |
| [ISO\_INSTANT](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_INSTANT) | Date and Time of an Instant | '2011-12-03T10:15:30Z' |
| [RFC\_1123\_DATE\_TIME](https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#RFC_1123_DATE_TIME) | RFC 1123 / RFC 822 | 'Tue, 3 Jun 2008 11:05:30 GMT' |

### Patterns for Formatting and Parsing

Symbol Meaning Presentation Examples

------ ------- ------------ -------

G era text AD; Anno Domini; A

u year year 2004; 04

y year-of-era year 2004; 04

D day-of-year number 189

M/L month-of-year number/text 7; 07; Jul; July; J

d day-of-month number 10

Q/q quarter-of-year number/text 3; 03; Q3; 3rd quarter

Y week-based-year year 1996; 96

w week-of-week-based-year number 27

W week-of-month number 4

E day-of-week text Tue; Tuesday; T

e/c localized day-of-week number/text 2; 02; Tue; Tuesday; T

F week-of-month number 3

a am-pm-of-day text PM

h clock-hour-of-am-pm (1-12) number 12

K hour-of-am-pm (0-11) number 0

k clock-hour-of-am-pm (1-24) number 0

H hour-of-day (0-23) number 0

m minute-of-hour number 30

s second-of-minute number 55

S fraction-of-second fraction 978

A milli-of-day number 1234

n nano-of-second number 987654321

N nano-of-day number 1234000000

V time-zone ID zone-id America/Los\_Angeles; Z; -08:30

z time-zone name zone-name Pacific Standard Time; PST

O localized zone-offset offset-O GMT+8; GMT+08:00; UTC-08:00;

X zone-offset 'Z' for zero offset-X Z; -08; -0830; -08:30; -083015; -08:30:15;

x zone-offset offset-x +0000; -08; -0830; -08:30; -083015; -08:30:15;

Z zone-offset offset-Z +0000; -0800; -08:00;

p pad next pad modifier 1

' escape for text delimiter

'' single quote literal '

[ optional section start

] optional section end

# reserved for future use

{ reserved for future use

} reserved for future use

# Liste déroulante

## Format des Dates

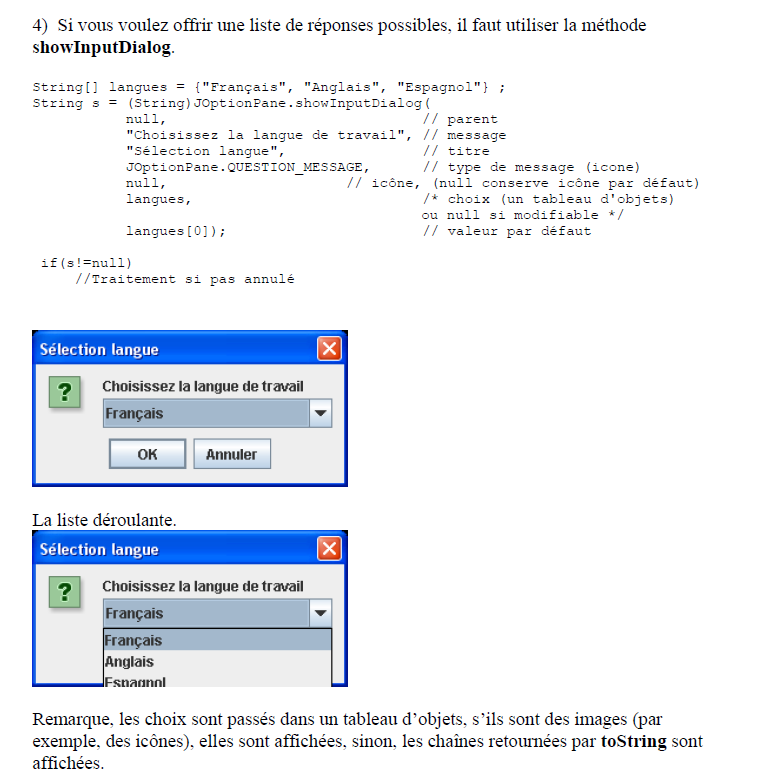
TU

<https://www.delftstack.com/fr/howto/java/how-to-get-the-current-date-time-in-java/>

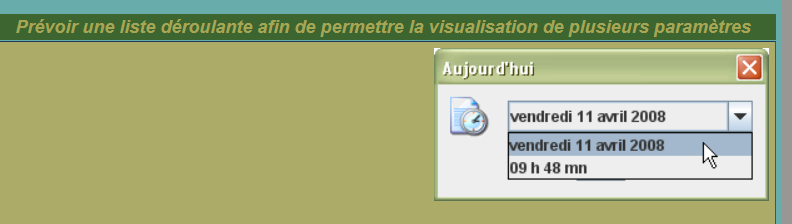
eu

usa

uk



[C:\Users\Djamel\Workspace\jSwing\JOptionPane\JOptionPane\WorkItems](file:///C:\Users\Djamel\Workspace\jSwing\JOptionPane\JOptionPane\WorkItems)

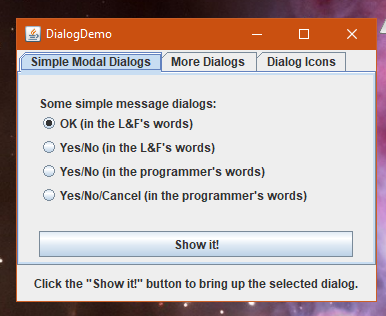


<http://remy-manu.no-ip.biz/Java/Tutoriels/IHM/dialogue.html>

# JTabbedPane

Une image contenant texte

Description générée automatiquement

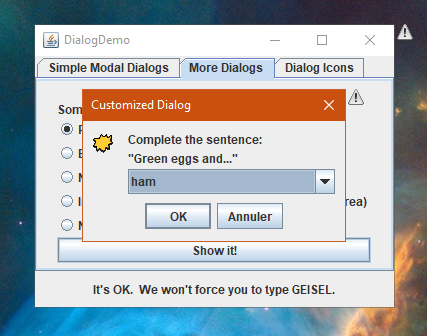


Une image contenant texte, capture d’écran, moniteur

Description générée automatiquement

Une image contenant texte

Description générée automatiquement



Une image contenant texte, arbre, capture d’écran

Description générée automatiquement

Une image contenant texte, capture d’écran, arbre

Description générée automatiquement

# Links

<https://www.delftstack.com/fr/howto/java/how-to-get-the-current-date-time-in-java/>

<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>

<https://www.delftstack.com/fr/howto/java/java-get-current-timestamp/>

<https://koor.fr/Java/API/java/lang/System/currentTimeMillis.wp>

1. <https://en.wikipedia.org/wiki/Time_zone> [↑](#footnote-ref-1)
2. <https://fr.wikipedia.org/wiki/Fuseau_horaire> [↑](#footnote-ref-2)
3. <https://docs.oracle.com/javase/8/docs/api/java/time/ZoneId.html> [↑](#footnote-ref-3)
4. <https://docs.oracle.com/javase/8/docs/api/java/time/ZoneId.html> [↑](#footnote-ref-4)
5. <https://devstory.net/13715/java-zoneid> [↑](#footnote-ref-5)