

# MIGRATING TO THE JAVA DATE TIME API

VERSION 8





federico-parezzan



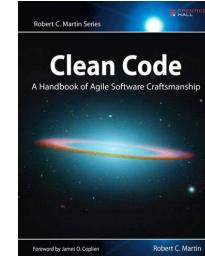
parez93

# Federico Parezzan

➤ **Java backend developer**



Dialogflow



Homo sum, humani nihil a me alienum puto ~ Terenzio

# Agenda

1. A look to the past
2. Java 8+ API
3. Demo
4. Database hints
5. Parsing and formatting hints
6. A little bit of infor-magic... 

## Date & Calendar

1. Poor API design
2. Date as time interval
3. No Immutability
4. No Thread-Safe
5. Formatting classes work only with Date

## Java 8+

1. API Clarity
2. Date as value
3. Immutability
4. Thread-Safe

# DEMO

# API

java.time	Meaning	Time zoned	Example
LocalDate	Date	NO	Current date
LocalTime	Time	NO	Current time
LocalDateTime	Date and time	NO	Start of a conference call in Italy
ZonedDateTime	Date and time	YES	Start of a conference call in America
Duration	Time between two instants	NO	Length of a meetup
Period	Amount of time in years, months and days	NO	Length of university studies
Instant	Specific moment in time	YES	Timestamp

# DEMO

LocalDate, LocalTime & LocalDateTime

# DEMO

## ZonedDateTime

# DEMO

Instant

# DEMO

Period & Duration

## Database types mapping

SQL	JAVA 8+
DATE	LocalDate
TIME	LocalTime
TIMESTAMP	LocalDateTime
TIME WITH TIME ZONE	OffsetTime
TIMESTAMP WITHOUT TIME ZONE	OffsetDateTime

# Parsing & Formatting

```
LocalDateTime dateTime = LocalDateTime.of(2020, 9, 1, 13, 14);
DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");
String formattedDateTime = dateTime.format(formatter);
System.out.println(formattedDateTime);
```

```
LocalDateTime parsedDate =
LocalDateTime.parse(formattedDateTime, formatter);
System.out.println(parsedDate);
```

# Cheatsheet

## Java Date Time in Java 8

```
java 1.0           java.util.Date;java.text.Format
java 1.2           java.util.Calendar
java 8             java.time.*  
java 8 classes: LocalDate, LocalTime, LocalDateTime, Month (1-12), Period, Duration
```

## Creating Dates and Times

of()	Overridden by LocalData, LocalTime and LocalDateTime
LocalDate.of(y,M,d);	M: int or Month(1-12)
LocalTime.of(h,m,s,ns)	ns: nano second
LocalDateTime.of(y,M,d,h,m,s,ns)	LocalDateTime.of(d,t)
now()	Overridden by LocalData, LocalTime and LocalDateTime
parse(s,f)	Overridden by LocalData, LocalTime and LocalDateTime
LocalDateTime.parse(s,f)	f must be DateTimeFormatter
LocalDate d= new LocalDate() //not compiled	
LocalDate d= LocalDate.of(2020,10,27); //not compile	

## Manipulating Dates and Times

plusYYY(int)	
minusYYY(int)	
return new immutable, original not changed, like new String()	
chain of method on date and time	
getXXX(); //XXX= DayOfWeek,DayOfYear, Month, Year	
dt1.isBefore(dt2)	
dt1.isAfter(dt2)	
YYY = Years / Months /Days/Hours/Minutes/Seconds/Nanos;	
Java 8 date & time are immutable, like String objects.	

## Period and Duration

java.time.Period; java.time.Duration;  
**4 ways to create a period**  
 Period.ofXXX(int); //XXX=Years,Months,Days  
 Period.of(int years,int months, int days);  
 Period.between(d1,d2)  
 Period.parse("P1Y2M3W4D");//same as Period.of(1, 2, 25)  
**Modify period**  
 plusYYY(int); //YYY = Days | Weeks | Months | Years  
 minusYYY(int); //YYY = Days | Weeks | Months | Years  
**Use period**  
 dt.plus/minus(period);  
 d.plus/minus(period);  
 time.plus/minus(duration); //if input period, UnsupportedOperationException  
 Chain of Periods method does not make sense

## Formatting

**2 methods to create a formatter**  
 DateTimeFormatter f=DateTimeFormatter.ofXXX(FormatStyle.SHORT|MEDIUM|LONG|FULL);  
 DateTimeFormatter f=DateTimeFormatter.ofPattern("yyy-MM-dd hh:mm:ss"); //M->1, MM->01, MMM->Jan, MMMM->January  
**Standard formatter:**  
 DateTimeFormatter.ISO\_LOCAL\_DATE/TIME/DATETIME  
 ISO\_LOCAL\_DATETIME: 2020-10-27T14:53:01  
**3 ways to use formatter**  
 f.format( date/time/datetime\_obj );  
 date/time/datetime\_obj.format( f );  
 LocalDate/Time/DateTime.parse(String s, DateTimeFormatter f);  
 XXX = LocalizedDate, LocalizedTime, LocalizedDateTime  
 date and time format legal for datatime object, datatime format not legal for date or time object

**GRAZIE PER L'ATTENZIONE**

