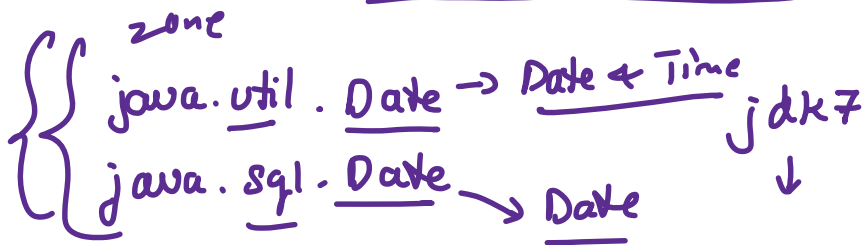


Date & Time APIs → java 8 features



↓
immutable
Thread Safe
= SimpleDateFormat → java.text

formatting & parsing

"DD/MM/YYYY HH:MM"

Date Time APIs

Immutable → Multi-Threading



→ java.time — { → LocalDate
→ LocalDateTime
→ LocalTime

"10-06-2020 HH:MM"
↓

→ java.time.format → formatting & parsing

→ java.time.temporal → First^{day} of a month
Last day of a month

→ java.time.zone

LocalDate

↓

java.time

→ import java.time.LocalDate;

a) public static LocalDate now() → yyyy-mm-dd

LocalDate today = LocalDate.now();

b) Creating LocalDate by providing input is yyyy-mm-dd

of

LocalDate.of(yyyy, mm, dd)

eg

LocalDate dt = LocalDate.of(2022, 10, 06)

S.O P(dt)

↓
Month: October

LocalDate dt = LocalDate.of(2022, Month.October, 06)

↓
java.time.DateTimeException

c) now(ZoneId)

LocalDate todayKolkata = LocalDate.now(ZoneId.of("Asia/Kolkata"))

```
Set<String>zonelds=ZoneId.getAvailableZoneIds();  
zonelds.stream().forEach(System.out::println);
```

java.time.zone.ZoneRulesException

Base date → 01/01/1970

d) LocalDate.EPOCH

e) LocalDate dt = LocalDate.ofEpochDate(365)

1971, 01, 01

9, 11, 11 → 01-01-2014

2014 → 01-01-2014
 100
 []

f) of YearDay

LocalDate dt = LocalDate.ofYearDay(2014, 100)

→ 01-01-2014, 100

[10-01-2014]

LocalDate (-, -, -)

ii) LocalTime

HH:MM:SS:SSSS

a) now()

public static LocalTime now()

e.g

LocalTime tm = LocalTime.now()
 s.o.p(tm)

b) of

public static LocalTime of(hh, mi, ss, ssss)

LocalTime tm = LocalTime.of(09, 50, 30, 40)

java.time.DateTimeException

i) of(hh, mi)

ii) of(hh, mi, ss)

iii) of(hh, mi, ss, ssss)

c) LocalTime localTime = LocalTime.now (ZoneId.of("Asia/Kolkata"));
s.o.p(localTime)

~~2~~

Zone Rules Exception

d) 01,01,1970 → 00:00:00.00

LocalTime localTime = LocalTime.ofSecondOfDay(10000);

LocalDateTime

- a) `LocalDateTime dt = LocalDateTime.now();`
- b) `LocalDateTime dt = LocalDateTime.of(LocalDate.now(), LocalTime.now());`
- c) `LocalDateTime dt = LocalDateTime.of(yyyy, mm, dd, hh, mi, ss, sec)`
`LocalDateTime.of(2022, Month.OCTOBER, 6, 10, 05, 30, 40)`
↓
DateTimeException

- d) `LocalDateTime dt = LocalDateTime.now(ZoneId.of("Asia/Kolkata"))`

APJ Utility methods

- a) `isLeapYear()`

```
LocalDate dt = LocalDate(2000, 01, 01);  
if (dt.isLeapYear())  
    s.o.p("Leap year"),  
else  
    s.o.p("Not a leap year")
```

- b) Compare two `LocalDate` for before or after

i) `isBefore`

ii) `isAfter`

```
LocalDate today = LocalDate.now();
```

```
boolean x = today.isBefore(LocalDate(2020, 10, 30));  
↓  
false
```

↓
false

today.isAfter(LocalDate(2020, 10, 30));

↙ ↓
2022, 10, 06 true

c) plus / minus

1) plusDays(d)		1) minusDays(d)
2) plusWeeks(w)		2) minusWeeks(w)
3) plusMonths(m)		3) minusMonths(m)

e.g

LocalDate dt = LocalDate.now();
LocalDate d2 = dt.plusMonths(12)

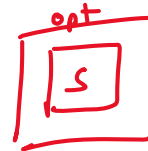
TemporalAdjuster.firstDayOfMonth(),
lastDayOfYear()

LocalDate today = LocalDate.now();

s.o.p("first date of this month: " + today.with(TemporalAdjuster.firstDayOfMonth()))

Optional → java.util

```
Student findById(Integer id)
{
    null
}
```



```
Student x = repo.findById(1001);
x.getRollno();
↳ NullPointerException
```



```
✓ Optional<Student> x;
x = repo.findById(1001);
if (x.isPresent())
    x.ifPresent(...)
```

Optional

Student findStudentByRollno(int rollno)

"SELECT * FROM Student WHERE rollno = " + rollno;

```
if (resultSet.next())
    return new Optional(s);
else
    return Optional.empty();
```

3

```
S = findStudentByRollno(1001)
S.o.p(S.getRollno())
```

if S.isPresent()

if (S.isEmpty())

↳ NullPointerException

1) empty()

Optional<Student> opt = Optional.empty();

2) of(T obj)

Optional<Student> opt = Optional.of(new Student(1001, "Amit", 55, 55));

Student s = fetchStudent();

```
if (s == null)
    return Optional.empty();
```



```

if (s == null)
    return Optional.empty();
else
    return Optional.of(s)

```

3) ofNullable(T obj)

```

Student s = fetchStudent();
return Optional.ofNullable(s);

```



1) Optional<Integer> opt = Optional.of(25)

2) Optional.of(new Student(1001, "A"))

3) T get() → 25
i = opt.get()

4) T nullable
↑ null



NoSuchElementException

Spring Data JPA

6) boolean isEmpty() opt.isEmpty()

7) boolean isPresent() if (opt.isPresent())

2

8) void ifPresent(Consumer<T> consumer)

```

opt.ifPresent(s → System.out.println(s));
opt.ifPresent(System.out::println)

```

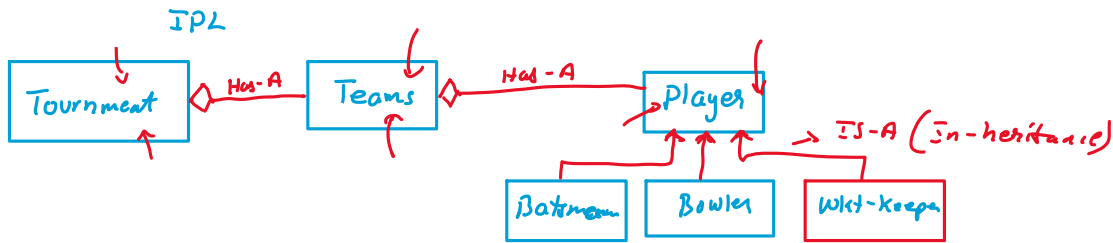
9) filter

```

opt.filter(s → if (s.getId() == 1001)
    return true;
else
    return false;
)
.ifPresent(System.out::println)

```

3)



```

class Players
{
    ~
    ~
    ~
}

class Batsman extends Player
{
    ~
    ~
    ~
}

class Bowler extends Player
{
    ~
    ~
    ~
}

class WktKeep extends Player
{
    ~
    ~
    ~
}
  
```

```

class Team
{
    String teamName;
    String cap;
    ~
    ~
    private ArrayList<Player> players;
    ~
    ~
}
  
```

```

class Tournament
{
    ~
    ~
    private String tournamentName;
    private String location;
    private ArrayList<Team> teams;
    ~
    ~
}
  
```

Tournament

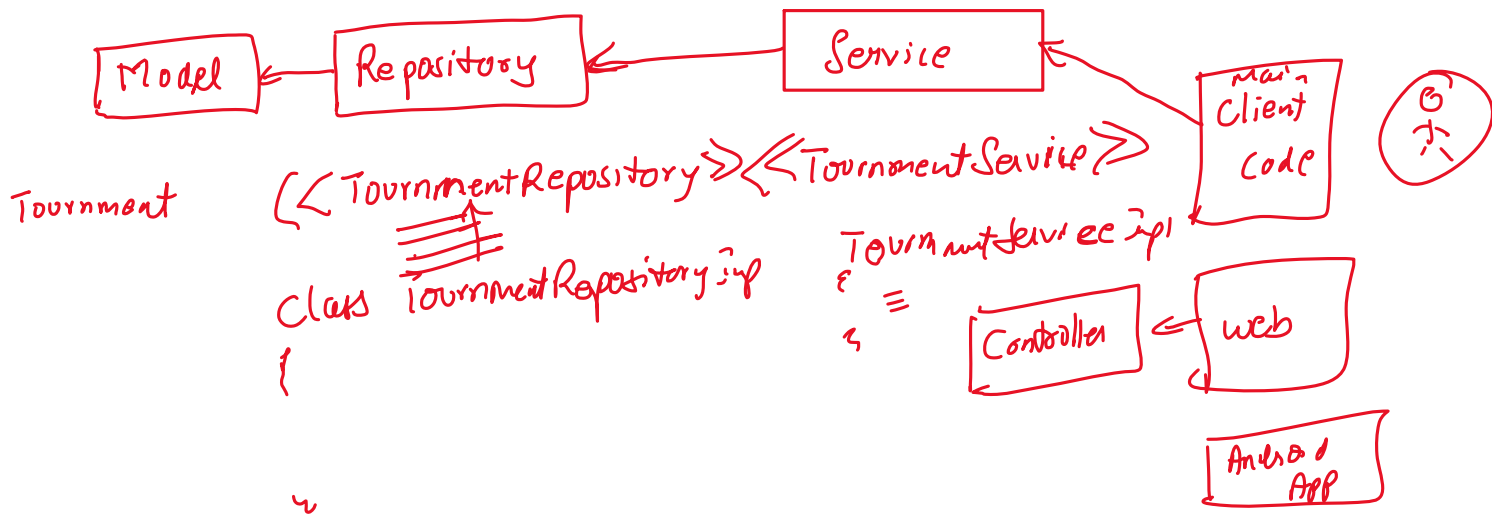
Teams

tournamentName : String
 location : String
 teams : ArrayList<Team>
 startDate : LocalDate
 endDate : LocalDate

Constructor
 Setter/Getter methods
 toString method
 equals & hashCode method
 addTeam() : void
 getTeams() : ArrayList<Team>
 getTeam(String teamName)

teamName : String
 ceo : String
 players : ArrayList<Player>

Constructor
 Setter/Getter methods
 toString
 equals & hashCode
 addPlayer() : void
 getPlayers() : ArrayList<Player>
 getPlayer(int playerId)



Interface → Repository, Service
 Class → Repository, Service, Model